

Proyecto Fin de Máster Ingeniería Aeronáutica

Modelado, simulación y control de un sistema de seguimiento de trayectorias para UAVs

Autor: Antonio José Aguilera Albendín

Tutor: Eduardo Fernández Camacho

Dpto. de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019



Proyecto Fin de Máster
Ingeniería Aeronáutica

Modelado, simulación y control de un sistema de seguimiento de trayectorias para UAVs

Autor:

Antonio José Aguilera Albendín

Tutor:

Eduardo Fernández Camacho

Catedrático

Dpto. de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019

Proyecto Fin de Máster: Modelado, simulación y control de un sistema de seguimiento de trayectorias para UAVs

Autor: Antonio José Aguilera Albendín
Tutor: Eduardo Fernández Camacho

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

La realización de este Trabajo Fin de Máster supone la finalización de mi etapa universitaria y es por eso que quisiera dar las gracias a todos aquellos que han hecho que sea posible. En primer lugar está claro a quiénes debo agradecer todo su esfuerzo y sacrificio: a mis padres, Antonio y M^a Carmen, ya que sin ellos no habría sido posible que hoy tuviera la formación que tengo y es algo que nunca podré terminar de agradecerse. Tampoco me puedo olvidar de mi familia, en especial de mi abuela Marina y de mi hermano, Juan Carlos.

En segundo lugar quiero agradecer a todos aquellos que me han ayudado cuando lo he necesitado: amigos, compañeros de clase, compañeros de piso, profesores. Ellos también han contribuido a que llegue este momento.

Finalmente quisiera darle las gracias a mi Tutor, Eduardo, sin el cual la realización de este Proyecto no habría sido factible.

*Antonio José Aguilera Albendín
Sevilla, 2019*

Resumen

Los drones o UAVs tienen un gran potencial en áreas muy diversas, ya que pueden desplazarse rápidamente sobre un terreno irregular o accidentado y superar cualquier tipo de obstáculo ofreciendo imágenes a vista de pájaro y otro tipo de información recogida por diferentes sensores. Su utilización ha crecido exponencialmente durante los últimos años debido a la gran cantidad de aplicaciones donde pueden ser muy útiles: búsqueda de personas desaparecidas, fotografía, video y cartografía aérea, prevención y control de incendios, agricultura, medio ambiente, construcción e inspecciones, control y análisis de multitudes, exploración de lugares de difícil acceso, etc.

Al tratarse de aeronaves autónomas es necesario dotarlas de un sistema de misión que se encargue de implementar la lógica necesaria para poder realizar la misión deseada. En la literatura podemos encontrar numerosas técnicas de control y guiado para un UAV: control mediante bucle cerrado (PID), control predictivo (MPC), neural network guidance, métodos de control y guiado directos, etc, teniendo cada uno sus ventajas e inconvenientes. En este proyecto se ha hecho uso del control PID debido a su buen funcionamiento y amplio uso en la industria. Además, debido a que el objetivo del proyecto no reside en la optimización de trayectorias, el control PID permite el guiado del UAV sin un alto gasto computacional.

En el proyecto desarrollado se modela el UAV X8 Skywalker, con configuración de ala volante. El modelo matemático utilizado para realizar la simulación consiste en un método dinámico no lineal que describe el comportamiento del UAV de la manera más realista posible. Se le dotará de un sistema de misión, basado en control mediante bucle cerrado, y diversos sensores útiles para diferentes tipos de aplicaciones.

Una vez implementado el sistema que modela el UAV X8 Skywalker se realizarán diversas simulaciones para estudiar el funcionamiento del sistema de misión implementado y su respuesta frente a perturbaciones atmosféricas. Además se mostrará cómo realizar la implementación con el simulador de vuelo FlightGear, software de código libre que permite ver en tiempo real la actuación de la aeronave durante la simulación.

Abstract

Drones or UAVs have great potential in very diverse areas, as they can move quickly over uneven or rugged terrain and overcome any obstacle offering bird's eye view and other information collected by different sensors. Its use has grown exponentially in recent years due to the large number of applications where they can be very useful: search for missing persons, photography, video and aerial mapping, prevention and control of fires, agriculture, environment, construction and inspections, control and analysis of crowds, exploration of places of difficult access, etc.

As autonomous aircrafts, it is necessary to provide them with a mission system that is responsible for implementing the logic necessary to carry out the desired mission. In the literature we can find many control and guidance techniques for an UAV: closed loop control (PID), predictive control (MPC), neural network guidance, direct control and guidance methods, etc., each having its advantages and disadvantages. In this project, PID control has been used due to its good operation and wide use in the industry. In addition, because the objective of the project does not reside in the trajectory optimization, the PID control allows the UAV to be guided without a high computational expense.

In the developed project, the UAV X8 Skywalker is modeled, with flying wing configuration. The mathematical model used to perform the simulation consists of a nonlinear dynamic method that describes the behavior of the UAV in the most realistic way possible. It will be equipped with a mission system, based on closed loop control, and various sensors useful for different types of applications.

Once the system that models the UAV X8 Skywalker is implemented, several simulations will be carried out to study the operation of the implemented mission system and its response to atmospheric disturbances. It will also show how to perform the implementation with the flight simulator FlightGear, free code software that allows to see in real time the performance of the aircraft during the simulation.

Índice

Agradecimientos	I
Resumen	III
Abstract	V
1 Introducción	1
1.1 Alcance del Proyecto	1
1.2 UAV de partida	1
1.3 Sistemas de referencia	2
2 Modelado	5
2.1 Cinemática y dinámica	5
2.1.1 Variables de estado	5
2.1.2 Cinemática	5
2.1.3 Dinámica	6
2.2 Fuerzas y momentos	7
2.2.1 Modelo gravitatorio	7
2.2.2 Modelo aerodinámico	8
Superficies de control	9
Aerodinámica longitudinal	10
Aerodinámica lateral	13
Parámetros del UAV	14
2.2.3 Modelo propulsivo	14
2.2.4 Perturbaciones atmosféricas	15
2.3 Resumen	16
2.4 Trimado UAV	17
3 Control: Diseño de Autopiloto	19
3.1 Autopiloto Longitudinal	19
3.1.1 Controlador de Cabeceo	20
3.1.2 Controlador de Altitud	21
3.1.3 Controlador de Velocidad mediante Cabeceo	22
3.1.4 Controlador de Velocidad mediante Empuje	23
3.1.5 Máquina de Estados de Control de Altitud	23
3.2 Autopiloto Lateral	24
3.2.1 Controlador de Alabeo	24
3.2.2 Controlador de Rumbo	25
4 Guiado: Sistema de Misión	27

4.1	Algoritmo para cálculo de ruta	27
4.2	Planificador de vuelo TSP	30
5	Sensores y EKF	33
5.1	Sensores a bordo	33
5.1.1	Acelerómetro	33
5.1.2	Giróscopo	33
5.1.3	Altímetro barométrico	34
5.1.4	Anemómetro	34
5.1.5	Magnetómetro	34
5.2	Sistema GPS	35
5.3	Filtro de Kalman Extendido	36
5.3.1	EKF Posición	37
5.3.2	EKF Actitud	38
6	Implementación en Matlab-Simulink	39
6.1	Bloque de Cinemática y Dinámica	39
6.1.1	Cinemática	39
6.1.2	Dinámica	40
6.2	Bloque de Fuerzas y Momentos	40
6.2.1	Modelo gravitatorio	40
6.2.2	Modelo aerodinámico	41
6.2.3	Viento	41
6.2.4	Fuerzas y Momentos	43
6.3	Controladores	43
6.3.1	Máquina de estados de Control de Altitud	43
6.3.2	Controlador de Cabeceo	43
6.3.3	Controlador de Altitud	44
6.3.4	Controlador de Velocidad mediante Cabeceo	45
6.3.5	Controlador de Velocidad mediante Empuje	45
6.3.6	Controlador de Alabeo	45
6.3.7	Controlador de Rumbo	46
6.4	Misión	46
6.5	Sensores	47
6.5.1	Acelerómetro	47
6.5.2	Giróscopo	47
6.5.3	Altímetro barométrico	47
6.5.4	Anemómetro	48
6.5.5	Magnetómetro	48
6.6	Sistema GPS	48
6.7	Filtro de Kalman Extendido	49
6.7.1	EKF Posición	49
6.7.2	EKF Actitud	50
7	Resultados	51
7.1	Misión realizada	51
7.2	Trayectoria	51
7.3	Actitud	53
7.4	Velocidad	55
7.5	Señal PWM de Empuje	55
7.6	Elevones	56
7.7	Coeficientes aerodinámicos	57
7.8	Acelerómetro	57
7.9	Giróscopo	58

7.10	Sistema GPS	58
8	Estudio de Robustez	61
8.1	Efecto de la intensidad del viento	61
8.2	Efecto de la dirección del viento	63
8.3	Efecto de la velocidad de vuelo	64
9	Simulador FlightGear	67
10	Conclusiones y Desarrollo Futuro	71
	Anexos	72
A	Representación en Google Eearth	75

1 Introducción

1.1 Alcance del Proyecto

El objetivo de este proyecto consiste en desarrollar un modelo matemático que describa el comportamiento dinámico del UAV Skywalker X8 y posteriormente estudiar diferentes misiones que pueda llevar a cabo. En el Capítulo 2 se desarrollan las ecuaciones que modelan el comportamiento de la aeronave, que corresponde a un sistema dinámico no lineal. Para ello haremos uso de los parámetros que modelan la aeronave presentes en [1].

Una vez que tenemos las ecuaciones que modelan la aeronave, se va a proveer al UAV de los sistemas necesarios para realizar la misión que nos interesa: un vuelo horizontal en el que de forma autónoma sea capaz de pasar por una serie de waypoints definidos. Para ello se llevará a cabo la implementación de un sistema autopiloto y de misión en los Capítulos 3 y 4. Además, en el Capítulo 5 se van a modelar diversos sensores que proporcionan datos de vuelo interesantes para poder implementar un Observador basado en el Filtro de Kalman. En los Capítulos 6 y 7 se muestra cómo se ha implementado dicho sistema en el software Matlab-Simulink y los resultados obtenidos durante la simulación, respectivamente.

En el Capítulo 8 se va a estudiar cómo afectan las perturbaciones atmosféricas debidas a viento en el comportamiento del UAV.

Finalmente, se realizará una implementación en el simulador de vuelo FlightGear y se abordarán las conclusiones que se obtienen a la finalización de la elaboración del proyecto, así como posibles vías de desarrollo futuro.

1.2 UAV de partida

La aeronave que se pretende simular en este proyecto se trata del *X8 Skywalker*. Corresponde a un UAV de tipo ala volante con las siguientes características físicas:

- Masa: $m = 3.797 \text{ kg}$
- Envergadura: $b = 2.10 \text{ m}$
- Cuerda: $c = 0.3571 \text{ m}$
- Longitud: $L = 0.78 \text{ m}$
- Superficie alar: $S = 0.75 \text{ m}^2$

las cuales han sido obtenidas de [1]. Al tratarse de un ala volante no dispone de timón de profundidad (rudder), por lo que el control de guiñada y balance se encuentra acoplado, tal y como se desarrollará en el Capítulo 3. Además al tener una masa tan pequeña, su comportamiento se verá condicionado por la intensidad y dirección del viento incidente, desarrollado en el Capítulo 8.



Figura 1.1 X8 Skywalker..

1.3 Sistemas de referencia

Para el desarrollo de las ecuaciones que definen la cinemática y dinámica de la aeronave es necesario hacer uso de diferentes sistemas de referencia debido a la gran cantidad de acciones a las que se encuentra sometida. Los sistemas empleados son:

- **Sistema de Ejes Tierra (ECEF, Earth Centered, Earth Fixed):** empleado cuando se quiere conocer la posición respecto de la Tierra, ya que es solidario con el movimiento de ella. Los parámetros empleados para conocer la posición en este sistema de ejes son la altitud (h), longitud (λ) y latitud (φ).

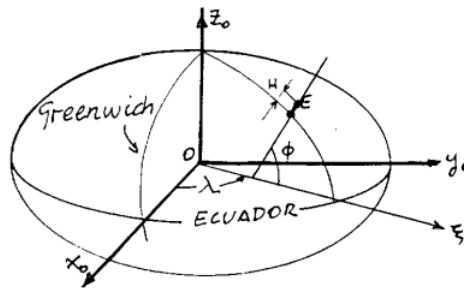


Figura 1.2 Sistema de coordenadas de Ejes Tierra..

- **Sistema de Ejes de Navegación (NED, North, East, Down):** sistema de horizonte local. Se encuentra referenciado respecto a un punto que se puede encontrar en la superficie terrestre o no (aeronave). Está definido por las direcciones Norte (N), Este (E) y vertical hacia abajo (D).
- **Sistema de Ejes Cuerpo (BFS, Body Fixed System):** sistema de referencia empleado para definir la actitud de la aeronave respecto al sistema de Ejes de Navegación. Caracterizado por los ángulos de Euler de alabeo (ϕ), cabeceo (θ) y guiñada (ψ).
- **Sistema de Ejes Viento (WF, Wind Frame):** empleado para orientar el viento respecto de la aeronave. Definido por el ángulo de ataque (α) y ángulo de resbalamiento (β).

A partir de las matrices de cosenos directores (DCM) es posible pasar de un sistema de referencia a otro. Multiplicando un vector expresado en un determinado sistema de referencia por la correspondiente matriz de cosenos directores queda expresado en el sistema de referencia que sea de interés.

$$V_b = R_a^b V_a$$

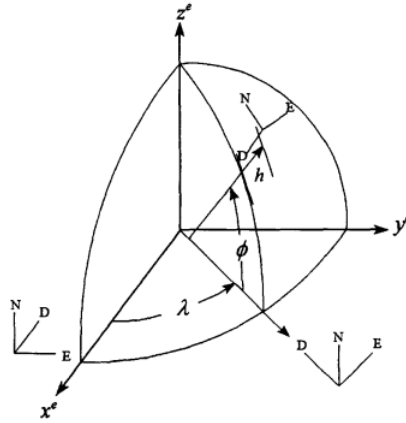


Figura 1.3 Sistema de Ejes de Navegación..

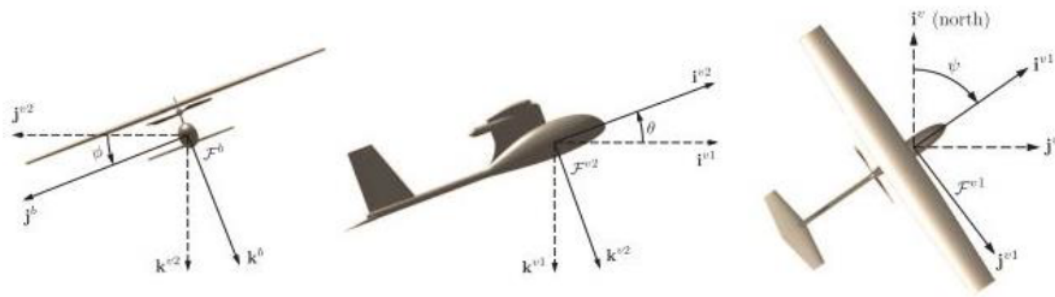


Figura 1.4 Sistema de Ejes Cuerpo..

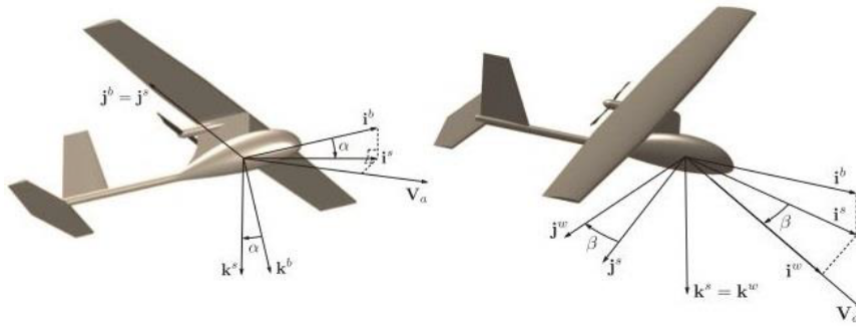


Figura 1.5 Sistema de Ejes Viento..

Una particularidad de estas matrices es que son ortogonales ($R_b^a = (R_a^b)^T$), lo cual permite transformar en sentido contrario multiplicando solamente por la matriz traspuesta

$$V_a = R_b^a V_b = (R_a^b)^T V_b \quad (1.1)$$

Es posible pasar de un sistema de referencia a otro a partir de sistemas de referencia intermedios, es decir, la matriz de cosenos directores que permite transformar un vector expresado en un determinado sistema de referencia en otro se puede expresar mediante el producto de matrices de cosenos directores intermedias

$$A \rightarrow B \rightarrow C \rightarrow D$$

$$V_d = R_a^d V_a = R_c^d R_b^c R_a^b V_a \quad (1.2)$$

Las matrices de cosenos directores que interesan para la consecución de este proyecto son aquellas que permiten pasar de un sistema de referencia a otro de los sistemas desarrollados anteriormente

- Matriz de rotación de ejes Tierra a ejes de navegación:

$$R_g^n = \begin{bmatrix} -\sin(\varphi) \cos(\lambda) & -\sin(\varphi) \sin(\lambda) & \cos(\varphi) \\ -\sin(\lambda) & \cos(\lambda) & 0 \\ -\cos(\varphi) \cos(\lambda) & -\cos(\varphi) \sin(\lambda) & -\sin(\varphi) \end{bmatrix} \quad (1.3)$$

- Matriz de rotación de ejes de navegación a ejes cuerpo:

$$R_n^b = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ -c_\phi s_\psi + s_\phi s_\theta c_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & s_\phi c_\theta \\ s_\phi s_\psi + c_\phi s_\theta c_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix} \quad (1.4)$$

- Matriz de rotación de ejes viento a ejes cuerpo:

$$R_w^b = \begin{bmatrix} \cos(\beta) \cos(\alpha) & -\sin(\beta) \cos(\alpha) & -\sin(\alpha) \\ \sin(\beta) & \cos(\beta) & 0 \\ \cos(\beta) \sin(\alpha) & -\sin(\beta) \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (1.5)$$

donde se ha empleado la notación $c_x \equiv \cos(x)$, $s_x \equiv \sin(x)$.

2 Modelado

2.1 Modelo cinemático y dinámico

En este apartado se desarrollarán las ecuaciones que modelan el comportamiento del X8. Primero se realizará un análisis de la cinemática que describe el movimiento de la aeronave y posteriormente se desarrollará la dinámica que produce dicho movimiento. Como se verá a continuación, se trata de un sistema dinámico no lineal, por lo que las variables de estado longitudinal y lateral de la aeronave se encontrarán acopladas, siendo necesario por tanto estudiar de forma conjunta la dinámica longitudinal y lateral.

2.1.1 Variables de estado

Las ecuaciones de movimiento para un UAV se describen mediante doce variables de estado. Hay tres estados de posición y tres estados de velocidad asociados con el movimiento de traslación del UAV. De manera similar, hay tres posiciones angulares y tres estados de velocidad angular asociados con el movimiento de rotación. Dichas variables se encuentran recogidas en la tabla 2.1.

Tabla 2.1 Variables de estado..

Símbolo	Descripción
P_n	Posición inercial del UAV respecto al norte en ejes NED.
P_e	Posición inercial del UAV respecto al este en ejes NED.
P_d	Posición inercial del UAV respecto al centro de la tierra en ejes NED.
u	Velocidad del UAV en dirección longitudinal en ejes Cuerpo.
v	Velocidad del UAV en dirección lateral en ejes Cuerpo.
w	Velocidad del UAV en dirección vertical en ejes Cuerpo.
ϕ	Ángulo de balance (roll).
θ	Ángulo de cabeceo (pitch).
ψ	Ángulo de alabeo (yaw).
P	Velocidad angular de balance.
Q	Velocidad angular de cabeceo.
R	Velocidad angular de guiñada.

2.1.2 Cinemática

El modelo cinemático desarrollado a continuación corresponde a las ecuaciones que permiten calcular la derivada del vector posición de la aeronave ($\dot{p}_n, \dot{p}_e, \dot{p}_d$), expresado en ejes de navegación. Dichas derivadas se calculan a partir de las componentes de la velocidad lineal (u, v, w) y su orientación (ϕ, θ, ψ), expresada mediante ángulos de Euler.

$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{bmatrix} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.1)$$

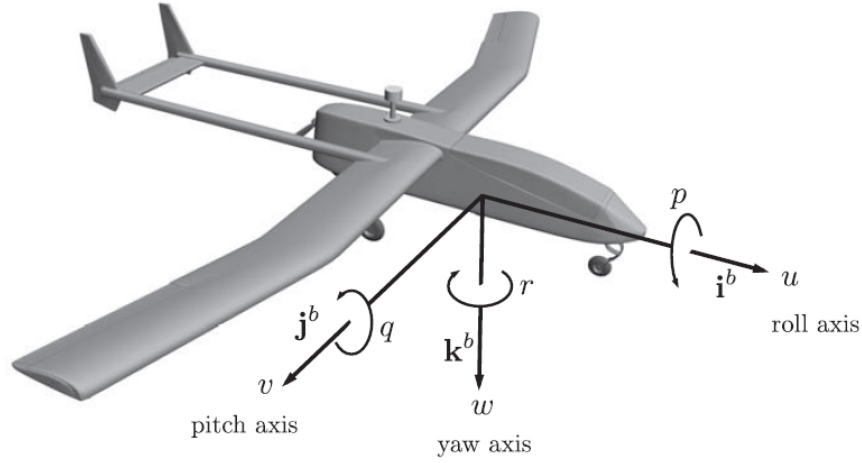


Figura 2.1 Variables de estado..

donde se ha empleado la notación $c_x \equiv \cos(x)$, $s_x \equiv \sin(x)$.

Respecto a los ángulos de Euler empleados en (2.1) para expresar los ángulos de alabeo (ϕ), cabeceo (θ) y guiñada (ψ), se pueden obtener sus derivadas ($\dot{\phi}, \dot{\theta}, \dot{\psi}$) a partir de las componentes de la velocidad angular (p, q, r).

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) \sec(\theta) & \cos(\phi) \sec(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.2)$$

Integrando los resultados obtenidos en (2.1) y (2.2) se obtienen la posición y actitud de la aeronave, respectivamente.

2.1.3 Dinámica

Para la obtención del modelo que representa la dinámica de la aeronave se va a considerar un modelo de Tierra plana, el cual es apropiado para vehículos pequeños, y a emplear la segunda Ley de Newton, para la cual es necesario tener un sistema de referencia inercial al que se encuentre asociado el movimiento de la aeronave.

Así, se puede obtener la aceleración lineal ($\dot{u}, \dot{v}, \dot{w}$) y angular ($\dot{p}, \dot{q}, \dot{r}$) de la aeronave, expresadas en ejes de navegación.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{m} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \quad (2.3)$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \Gamma_1 pq - \Gamma_2 qr \\ \Gamma_5 pr - \Gamma_6 (p^2 - r^2) \\ \Gamma_7 pq - \Gamma_1 qr \end{bmatrix} + \begin{bmatrix} \Gamma_3 l + \Gamma_4 n \\ \frac{m}{I_y} \\ \Gamma_4 l + \Gamma_8 n \end{bmatrix} \quad (2.4)$$

donde

$$\begin{aligned}
 \Gamma_1 &= \frac{I_{xz}(I_x - I_y + I_z)}{\Gamma} \\
 \Gamma_2 &= \frac{I_z(I_z - I_y) + I_{xz}^2}{\Gamma} \\
 \Gamma_3 &= \frac{I_z}{\Gamma} \\
 \Gamma_4 &= \frac{I_{xz}}{\Gamma} \\
 \Gamma_5 &= \frac{I_z - I_x}{I_y} \\
 \Gamma_6 &= \frac{I_{xz}}{I_y} \\
 \Gamma_7 &= \frac{I_x(I_x - I_y) + I_{xz}^2}{\Gamma} \\
 \Gamma_8 &= \frac{I_x}{\Gamma}
 \end{aligned} \tag{2.5}$$

siendo

$$\Gamma = I_x I_z - I_{xz}^2$$

2.2 Fuerzas y momentos

En esta sección se van a desarrollar las hipótesis y ecuaciones que dan lugar al modelo de fuerzas y momentos que experimenta la aeronave, que dan lugar al cierre del sistema que modela nuestro UAV.

Para modelar la fuerza total que experimenta la aeronave se puede hacer uso del Principio de Superposición y descomponer la fuerza resultante como la suma de tres componentes: fuerza gravitatoria, aerodinámica y propulsiva.

$$\sum \vec{F} = \vec{F}_{gravitatoria} + \vec{F}_{aerodinámica} + \vec{F}_{propulsiva} \tag{2.6}$$

Para el momento resultante que experimenta la aeronave se vuelve a hacer uso de nuevo del Principio de Superposición, considerando el momento total como el producido por la suma del momento producido por la fuerza aerodinámico y propulsivo. En este caso, la fuerza gravitatoria no produce momento en la aeronave, ya que ésta se considera como una fuerza puntual aplicada en el centro de gravedad de la aeronave.

$$\sum \vec{M} = \vec{M}_{aerodinámico} + \vec{M}_{propulsivo} \tag{2.7}$$

2.2.1 Modelo gravitatorio

Para modelar el gradiente gravitatorio al que se encuentra sometida la aeronave se va a hacer uso del modelo de gravedad *Earth Gravitational Model 96 (EGM96)*, el cual se encuentra incluido en el sistema de coordenadas *World Geodetic System 84 (WGS84)*. Dicho modelo proporciona un valor del gradiente gravitatorio en función de la latitud (φ) y altitud de vuelo.

Para el modelo matemático empleado por el *EGM96* se ha hecho uso del empleado en [3]

$$g(r, \varphi) = -\frac{GM_e}{r^2} \left[1 - \frac{3I_y a^2}{2r^2} (3 \sin^2(\varphi) - 1) \right] + \omega_e^2 r \cos^2(\varphi) \tag{2.8}$$

siendo G la constante de gravitación universal, M_e la masa del esferoide por el que se aproxima la Tierra, a el semieje mayor y ω_e la velocidad angular de la Tierra.

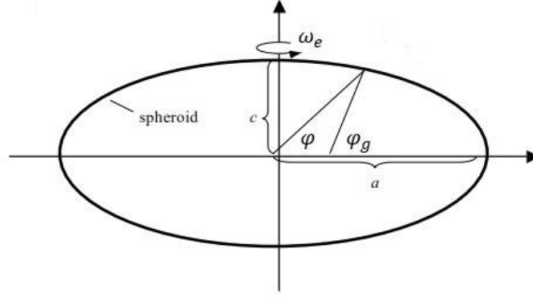


Figura 2.2 Esferoide del sistema WGS84..

Considerando que la fuerza gravitatoria se encuentra aplicada en el centro de masas de la aeronave y expresándola en ejes de navegación, se obtiene el siguiente vector de fuerza gravitatoria

$$\vec{F}_g = \begin{bmatrix} -mg \sin(\theta) \\ mg \cos(\theta) \sin(\phi) \\ mg \cos(\theta) \cos(\phi) \end{bmatrix} \quad (2.9)$$

Como ya se comentó anteriormente, debido a que la fuerza gravitatoria se encuentra aplicada en el centro de gravedad de la aeronave, ésta no produce momento.

2.2.2 Modelo aerodinámico

Para el desarrollo del modelo aerodinámico de la aeronave se va a considerar que la fuerzas de presión distribuidas a lo largo del perfil del ala van a estar aplicadas como una fuerza puntual en el centro aerodinámico de la aeronave, dando lugar así a la fuerza de sustentación (\vec{F}_{lift}), resistencia aerodinámica (\vec{F}_{drag}) y el momento aerodinámicos (\vec{M}_a).

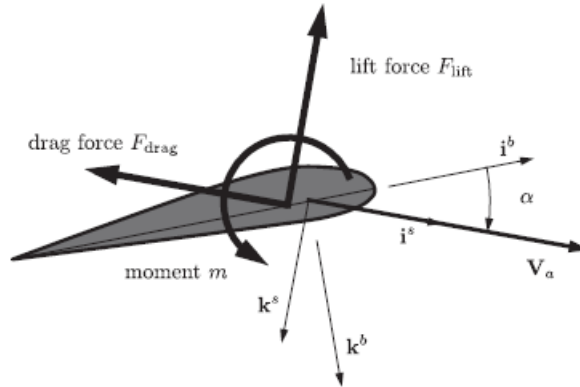


Figura 2.3 Fuerzas y momento aerodinámicos aplicadas en el centro aerodinámico del perfil..

Dichas fuerzas se modelan mediante las siguientes expresiones

$$\vec{F}_{lift} = \frac{1}{2} \rho \vec{V}_a^2 S C_L \quad (2.10)$$

$$\vec{F}_{drag} = \frac{1}{2} \rho \vec{V}_a^2 S C_D \quad (2.11)$$

$$\vec{m} = \frac{1}{2} \rho \vec{V}_a^2 S c C_m \quad (2.12)$$

donde C_L , C_D y C_m son los coeficientes aerodinámicos, S la superficie alar y c la cuerda media del perfil de

la aeronave. Los coeficientes aerodinámicos dependen de las superficies de control de la aeronave: alerones, elevadores y timón de profundidad, que su configuración dependerá del tipo de aeronave estudiada.

Superficies de control

Antes de desarrollar el modelo que define los coeficientes aerodinámicos de la aeronave en función de las superficies de control, se va a hacer una pequeña introducción sobre diferentes configuraciones de la aeronave.

Configuración estándar

La configuración estándar de una aeronave se puede ver en la Fig. 2.4, en la cual actúan como superficies de control los alerones (δ_a), elevadores (δ_e) y timón de profundidad (δ_r). La deflexión del alerón se puede considerar como una deflexión compuesta formada por

$$\delta_a = \frac{1}{2} (\delta_{a,left} - \delta_{a,right}) \quad (2.13)$$

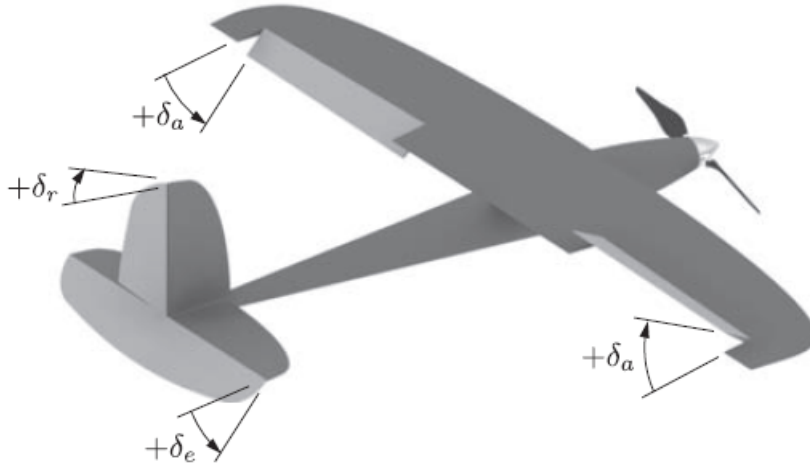


Figura 2.4 Variables de control aerodinámicas en configuración estándar..

Para aeronaves pequeñas existen dos configuraciones alternativas: cola en V y ala volante.

Configuración con cola en V

Este tipo de configuración sustituye el timón de profundidad (rudder) y elevadores (elevators) por los “ruddervators”. La deflexión angular del “ruddervator” derecho se denota como (δ_{rr}), y la deflexión angular del “ruddervator” izquierdo se denota como (δ_{rl}). Deflectarlos en distinto sentido produce el mismo efecto que el timón de profundidad, mientras que deflectarlos en el mismo sentido produce igual efecto que los elevadores. Matemáticamente se puede convertir entre “ruddervators” y timón-elevadores mediante

$$\begin{bmatrix} \delta_e \\ \delta_r \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \delta_{rr} \\ \delta_{rl} \end{bmatrix} \quad (2.14)$$

Ala volante

En este tipo de configuración (Fig. 2.6) los coeficientes aerodinámicos dependen de los elevones. La deflexión angular del elevón derecho se denota como δ_{er} , y la deflexión angular del elevón izquierdo se denota como δ_{el} . Deflectar los elevones en distinto sentido tiene el mismo efecto que los alerones (δ_a), mientras que deflectar los elevones juntos, tiene el mismo efecto que los elevadores (δ_e).

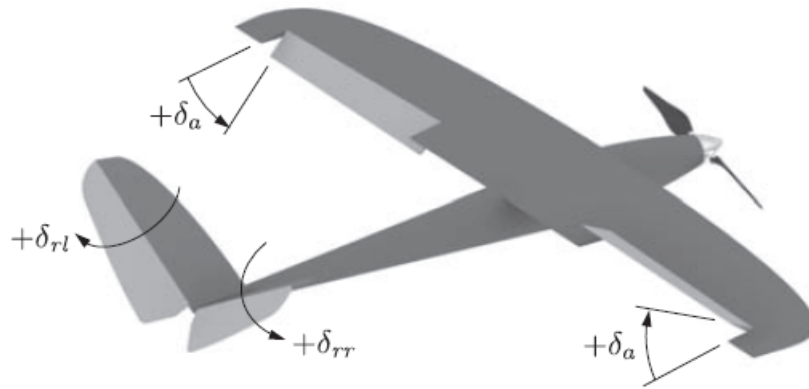


Figura 2.5 Variables de control aerodinámicas para cola en V..

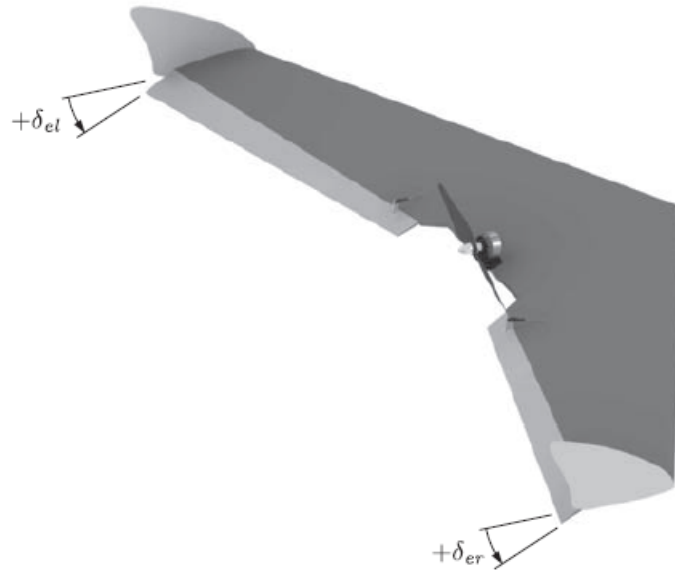


Figura 2.6 Variables de control aerodinámicas del ala volante..

Matemáticamente se puede convertir entre elevones y alerones-elevadores mediante

$$\begin{bmatrix} \delta_e \\ \delta_a \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \delta_{er} \\ \delta_{el} \end{bmatrix} \quad (2.15)$$

Debido a que es posible descomponer la aerodinámica de la aeronave en modelo longitudinal y lateral, se va a realizar dicha separación para llevar a cabo el estudio de los coeficientes aerodinámicos en función de las variables longitudinales y laterales, respectivamente.

Aerodinámica longitudinal

Las fuerzas y momentos aerodinámicos que afectan de manera longitudinal a la aeronave son aquellos contenidos en el plano de cabeceo: sustentación (*Lift*), resistencia aerodinámica (*Drag*) y momento de cabeceo (*pitching moment*). Dependen del ángulo de ataque (α), velocidad angular de cabeceo (q) y de los elevadores (δ_e).

$$F_{lift} = \frac{1}{2} \rho V_a^2 S C_L(\alpha, q, \delta_e) \quad (2.16)$$

$$F_{drag} = \frac{1}{2} \rho V_a^2 S C_D(\alpha, q, \delta_e) \quad (2.17)$$

$$m = \frac{1}{2} \rho V_a^2 S c C_m(\alpha, q, \delta_e) \quad (2.18)$$

Debido a que estas fuerzas y momento no son lineales, se suele emplear aproximaciones para calcularlas. Tomando series de Taylor de primer orden se obtienen las siguientes expresiones

$$F_{lift} = \frac{1}{2} \rho V_a^2 S \left[C_{L_0} + C_{L_\alpha} \alpha + C_{L_q} \frac{c}{2V_a} q + C_{L_{\delta_e}} \delta_e \right] \quad (2.19)$$

$$F_{drag} = \frac{1}{2} \rho V_a^2 S \left[C_{D_0} + C_{D_\alpha} \alpha + C_{D_q} \frac{c}{2V_a} q + C_{D_{\delta_e}} \delta_e \right] \quad (2.20)$$

$$m = \frac{1}{2} \rho V_a^2 S c \left[C_{m_0} + C_{m_\alpha} \alpha + C_{m_q} \frac{c}{2V_a} q + C_{m_{\delta_e}} \delta_e \right] \quad (2.21)$$

siendo C_{L_0} , C_{D_0} y C_{m_0} el valor de C_L , C_D y C_m cuando $\alpha = q = \delta_e = 0$, respectivamente y

$$C_{L_\alpha} = \frac{\partial C_L}{\partial \alpha}; C_{D_\alpha} = \frac{\partial C_D}{\partial \alpha}; C_{m_\alpha} = \frac{\partial C_m}{\partial \alpha}$$

$$C_{L_q} = \frac{\partial C_L}{\partial \frac{qc}{2V_a}}; C_{D_q} = \frac{\partial C_D}{\partial \frac{qc}{2V_a}}; C_{m_q} = \frac{\partial C_m}{\partial \frac{qc}{2V_a}}$$

$$C_{L_{\delta_e}} = \frac{\partial C_L}{\partial \delta_e}; C_{D_{\delta_e}} = \frac{\partial C_D}{\partial \delta_e}; C_{m_{\delta_e}} = \frac{\partial C_m}{\partial \delta_e}$$

Las ecuaciones (2.19), (2.20) y (2.21) son utilizadas normalmente para el diseño del modelo longitudinal de una aeronave que vuela con ángulo de ataque relativamente pequeño. Debido al carácter cuasi-estático de las fuerzas y momentos aerodinámicos las expresiones anteriores dan lugar a resultados satisfactorios. En cambio, los diseñadores de UAV deben tener en cuenta maniobras que dan lugar a fuerzas y momentos inestables, como pueden ser maniobras de alto ángulo de ataque y alta velocidad angular. También es importante señalar que el modelo anterior no tiene en cuenta los efectos de entrada en pérdida debido a superar el ángulo de entrada en pérdida (α_{stall}), a partir del cual el valor de la fuerza de sustentación disminuye bruscamente debido al desprendimiento de la capa límite del perfil.

Para incorporar dichos efectos en el modelo aerodinámico de la aeronave, se modifican ligeramente las ecuaciones (2.19) y (2.20) para que la sustentación y resistencia no sean lineales con el ángulo de ataque. Así se obtienen las expresiones

$$F_{lift} = \frac{1}{2} \rho V_a^2 S \left[C_L(\alpha) + C_{L_q} \frac{c}{2V_a} q + C_{L_{\delta_e}} \delta_e \right] \quad (2.22)$$

$$F_{drag} = \frac{1}{2} \rho V_a^2 S \left[C_D(\alpha) + C_{D_q} \frac{c}{2V_a} q + C_{D_{\delta_e}} \delta_e \right] \quad (2.23)$$

donde ahora C_L y C_D son funciones no lineales de α . Para ángulos de ataque superiores al de entrada en pérdida, el ala actúa principalmente como una placa plana, cuyo coeficiente se puede calcular según [4] como

$$C_{L, \text{placa plana}} = 2 \text{sign}(\alpha) \sin^2(\alpha) \cos(\alpha) \quad (2.24)$$

Un modelo de sustentación que incorpora el comportamiento de sustentación lineal común y los efectos de la entrada en pérdida viene dado por

$$C_L(\alpha) = (1 - \sigma(\alpha)) [C_{L_0} + C_{L_\alpha} \alpha] + \sigma(\alpha) [2 \text{sign}(\alpha) \sin^2(\alpha) \cos(\alpha)] \quad (2.25)$$

donde

$$\sigma(\alpha) = \frac{1 + e^{-M(\alpha - \alpha_0)} + e^{M(\alpha - \alpha_0)}}{(1 + e^{-M(\alpha - \alpha_0)})(1 + e^{M(\alpha - \alpha_0)})} \quad (2.26)$$

donde M y α_0 son constantes positivas. La función sigmoide (2.26) depende únicamente del ángulo de ataque y define la transición suave entre comportamiento lineal del ala al de placa plana. Dicha transición se encuentra representada en la Fig. (2.7).

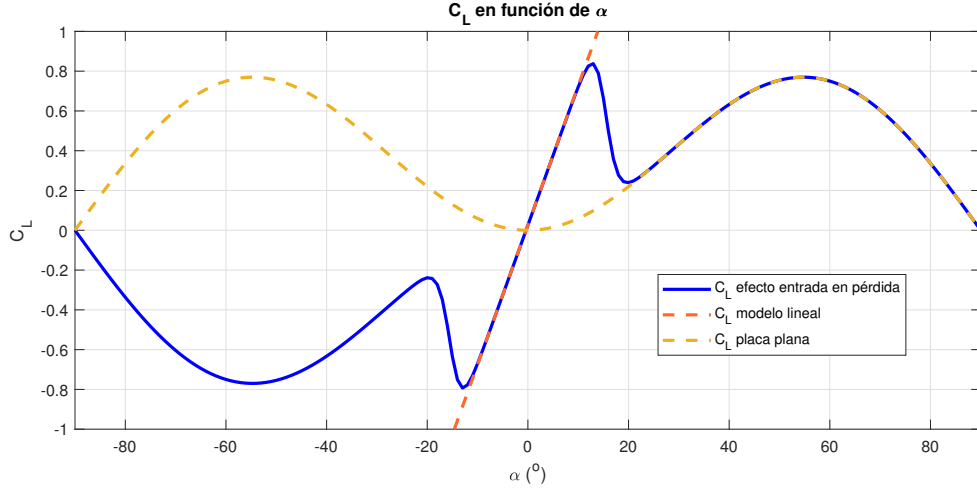


Figura 2.7 C_L frente a α .

Para aeronaves pequeñas, el coeficiente de sustentación lineal se puede aproximar razonablemente como

$$C_{L\alpha} = \frac{\pi AR}{1 + \sqrt{1 + (AR/2)^2}} \quad (2.27)$$

donde $AR = b^2/S$ es el alargamiento (*Aspect Ratio*), b la envergadura y S la superficie alar.

El coeficiente de resistencia C_D también es una función no lineal del ángulo de ataque. Hay dos contribuciones al coeficiente de resistencia, a saber: resistencia inducida y parásita [4]. La resistencia parasitaria, generada por el esfuerzo cortante del aire que se mueve sobre el ala y otros efectos, es aproximadamente constante y se denota por C_{D_p} . Para pequeños ángulos de ataque, la resistencia inducida es proporcional al cuadrado de la fuerza de sustentación. Combinando la resistencia parasitaria y la resistencia inducida, se obtiene

$$C_D(\alpha) = C_{D_p} + \frac{(C_{L_0} + C_{L\alpha}\alpha)^2}{\pi e AR} \quad (2.28)$$

El parámetro e corresponde al factor de eficiencia de Oswald, que se encuentra entre 0.8 y 1.0 [5]. Se puede ver una comparativa entre el modelo de resistencia lineal y cuadrática en la Fig. (2.8).

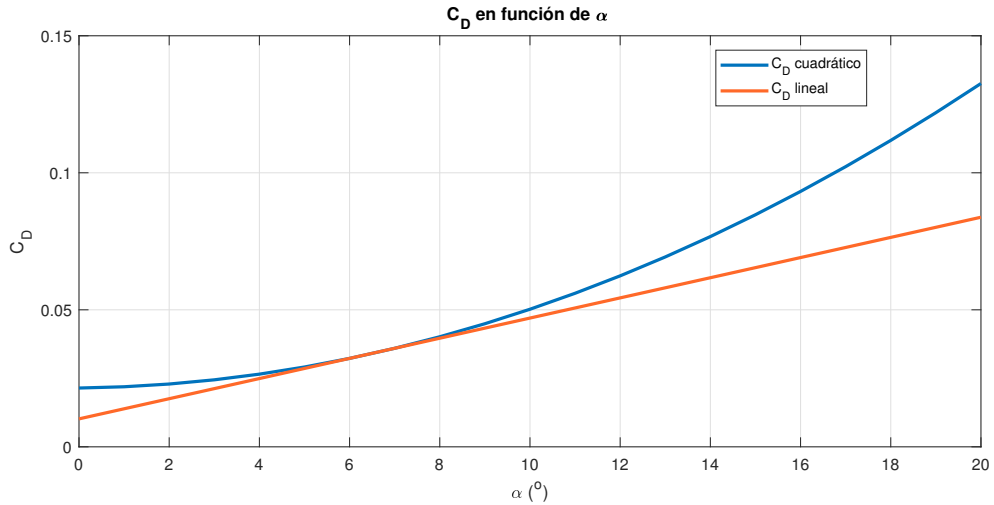


Figura 2.8 C_D frente a α . Modelos lineal y cuadrático..

Una vez que se ha obtenido un modelo para la fuerza de sustentación (2.22) y de resistencia (2.23) hay que expresar dichas fuerzas en ejes cuerpo,

$$\begin{aligned} \begin{bmatrix} f_x \\ f_z \end{bmatrix} &= \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} -F_{drag} \\ -F_{lift} \end{bmatrix} = \\ &= \frac{1}{2} \rho V_a^2 S \begin{bmatrix} -C_D(\alpha) c_\alpha + C_L(\alpha) s_\alpha + \frac{c}{2V_a} q \left(-C_{Dq} c_\alpha + C_{Lq} s_\alpha \right) + \delta_e \left(-C_{D\delta_e} c_\alpha + C_{L\delta_e} s_\alpha \right) \\ -C_D(\alpha) s_\alpha + C_L(\alpha) c_\alpha + \frac{c}{2V_a} q \left(-C_{Dq} s_\alpha - C_{Lq} c_\alpha \right) + \delta_e \left(-C_{D\delta_e} s_\alpha - C_{L\delta_e} c_\alpha \right) \end{bmatrix} \end{aligned} \quad (2.29)$$

siendo $c_\alpha \equiv \cos(\alpha)$ y $s_\alpha \equiv \sin(\alpha)$.

Las funciones empleadas en (2.22) y (2.23) modelan los coeficientes de sustentación (C_L) y resistencia (C_D) como funciones no lineales, que son válidas en un amplio rango de ángulos de ataque. Sin embargo, si se desean emplear modelos más sencillos, se pueden emplear los siguientes modelos lineales

$$C_L(\alpha) = C_{L_0} + C_{L_\alpha} \alpha \quad (2.30)$$

$$C_D(\alpha) = C_{D_0} + C_{D_\alpha} \alpha \quad (2.31)$$

En las figuras (2.7) y (2.8) se encuentra representados los dos modelos mencionados, lineal y no lineal para C_L y C_D , respectivamente.

El momento de cabeceo de la aeronave es generalmente una función no lineal del ángulo de ataque y debe determinarse mediante un túnel de viento o experimentos de vuelo para la aeronave específica de interés. A efectos de simulación, se utilizará el modelo lineal

$$C_m(\alpha) = C_{m_0} + C_{m_\alpha} \alpha \quad (2.32)$$

donde $C_{m_\alpha} < 0$ implica que la aeronave es estable frente a momentos de cabeceo.

Aerodinámica lateral

La fuerza aerodinámica que afecta a la dinámica lateral de la aeronave es aquella que se encuentra contenida a lo largo del eje y (ejes cuerpo), mientras que los momentos aerodinámicos corresponden a aquellos que producen un momento de alabeo (ϕ) y guiñada (ψ). La aerodinámica lateral está principalmente influenciada por el ángulo de resbalamiento (β). También están influenciados por la velocidad de balanceo (p), la velocidad de giro (r), la deflexión de los alerones (δ_a) y del timón de profundidad (δ_r).

Denotando la fuerza lateral como f_y y los momentos de balanceo y guiñada como l y n , respectivamente, tenemos

$$f_y = \frac{1}{2} \rho V_a^2 S C_Y(\beta, p, r, \delta_a, \delta_r) \quad (2.33)$$

$$l = \frac{1}{2} \rho V_a^2 S b C_l(\beta, p, r, \delta_a, \delta_r) \quad (2.34)$$

$$n = \frac{1}{2} \rho V_a^2 S b C_n(\beta, p, r, \delta_a, \delta_r) \quad (2.35)$$

donde C_Y , C_l y C_n son los coeficientes aerodinámicos adimensionales laterales, y b la envergadura de la aeronave.

Tal y como ocurría con el modelo aerodinámico longitudinal, estos coeficientes no son lineales en su relación con β , p , r , δ_a y δ_r . Sin embargo, en este caso los modelos lineales sí presentan resultados más precisos que en el caso longitudinal, por tanto, tal y como se hizo en las ecuaciones (2.19)-(2.21) se emplean aproximaciones mediante series de Taylor de primer orden, obteniéndose

$$f_y = \frac{1}{2} \rho V_a^2 S \left[C_{Y_0} + C_{Y_\beta} \beta + C_{Y_p} \frac{b}{2V_a} p + C_{Y_r} \frac{b}{2V_a} r + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r \right] \quad (2.36)$$

$$l = \frac{1}{2} \rho V_a^2 S b \left[C_{l_0} + C_{l_\beta} \beta + C_{l_p} \frac{b}{2V_a} p + C_{l_r} \frac{b}{2V_a} r + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r \right] \quad (2.37)$$

$$n = \frac{1}{2} \rho V_a^2 S b \left[C_{n_0} + C_{n_\beta} \beta + C_{n_p} \frac{b}{2V_a} p + C_{n_r} \frac{b}{2V_a} r + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r \right] \quad (2.38)$$

Debido a que estas fuerza y momentos se encuentran expresadas ya en ejes cuerpo, no es necesario aplicar matriz de rotación. El coeficiente C_{Y_0} es el valor del coeficiente C_Y cuando $\beta = p = r = \delta_a = \delta_r = 0$. En el caso de que la aeronave sea simétrica respecto del plano XZ, C_{Y_0} suele ser cero. Para aeronaves simétricas, los coeficientes C_{l_0} y C_{n_0} suelen ser típicamente cero.

Parámetros del UAV

En la tabla 2.2 se recogen los parámetros que caracterizan al UAV utilizados durante la elaboración del proyecto obtenidos de [1] y [2].

Tabla 2.2 Parámetros UAV Skywalker X8..

Parámetro	Valor	Parámetro	Valor
m (kg)	3.7970	C_{Y_0}	$3.2049 \cdot 10^{-18}$
b (m)	2.1000	C_{Y_β}	-0.1949
c (m)	0.3571	C_{Y_p}	-0.1172
S (m ²)	0.7500	C_{Y_r}	0.0959
AR	5.8800	$C_{Y_{\delta_a}}$	-0.0696
I_x (kg · m ²)	1.2290	C_{l_0}	$1.1518 \cdot 10^{-18}$
I_y (kg · m ²)	0.1702	C_{l_β}	-0.0765
I_z (kg · m ²)	0.8808	C_{l_p}	-0.4018
I_{xz} (kg · m ²)	0.9343	C_{l_r}	0.0250
e	0.9935	$C_{l_{\delta_a}}$	0.2987
α_0 (°)	15.2980	C_{m_0}	0.0180
C_{L_0}	0.0254	C_{m_α}	-0.2524
C_{L_α}	4.0191	$C_{m_{fp}}$	-0.2168
C_{L_q}	3.8954	$C_{m_{fp}}$	-1.3047
$C_{L_{\delta_e}}$	0.5872	$C_{m_{\delta_e}}$	-0.4857
C_{D_0}	0.0102	C_{n_0}	$-2.2667 \cdot 10^{-7}$
$C_{D_{\beta_1}}$	$-2.0864 \cdot 10^{-7}$	C_{n_β}	0.0403
$C_{D_{\beta_2}}$	0.0671	C_{n_p}	-0.0247
C_{D_q}	0	C_{n_r}	-0.1252
$C_{D_{\delta_e}}$	0.8461	$C_{n_{\delta_a}}$	0.0076
$C_{D_{\delta_e}}$	0.8461	$C_{n_{\delta_a}}$	0.0076

2.2.3 Modelo propulsivo

La expresión que modela la fuerza propulsiva viene dada por

$$F_{propulsiva} = \frac{1}{2} \rho S_{prop} C_{prop} \begin{bmatrix} (k_{motor} \delta_t)^2 - V_a^2 \\ 0 \\ 0 \end{bmatrix} \quad (2.39)$$

Una vez obtenida la fuerza propulsiva, se modela el empuje mediante una señal de modulación de ancho de pulso (PWM) δ_t , que actuará como variable de control del sistema. δ_t se encontrará acotada en el rango [0,1], valiendo 0 en el caso de no tener empuje (vuelo de descenso) y 1 siendo el empuje máximo que puede proporcionar la planta propulsiva (vuelo de ascenso).

Debido a que el empuje actúa a lo largo del eje longitudinal, éste no produce momento sobre la aeronave.

El aire que pasa a través de la hélice de la aeronave genera un par alrededor del eje de rotación de la hélice. El par aplicado por el motor a la hélice (y luego al aire) da como resultado un par igual y opuesto aplicado por la hélice al motor que se fija al cuerpo de la aeronave. Este par es opuesto a la dirección de rotación de la

hélice y proporcional al cuadrado de la velocidad angular de la hélice, según lo expresado por

$$T_p = -k_{T_p} (k_\Omega \delta_t)^2 \quad (2.40)$$

donde $\Omega = k_\Omega \delta_t$ es la velocidad de la hélice y k_{T_p} es una constante determinada experimentalmente. Los momentos debidos al sistema de propulsión son por tanto

$$M_{propulsivo} = \begin{bmatrix} -k_{T_p} (k_\Omega \delta_t)^2 \\ 0 \\ 0 \end{bmatrix} \quad (2.41)$$

Debido a que el momento producido por la hélice suele ser de poco efecto se ha decidido despreciarlo en la simulación realizada con el objeto de facilitar su implementación. Los parámetros empleados para modelar la fuerza y momento propulsivos han sido extraídos del Apéndice E de [2].

2.2.4 Perturbaciones atmosféricas

En esta sección, analizaremos las perturbaciones atmosféricas, como el viento, y describiremos cómo estas perturbaciones afectan a la dinámica de la aeronave. En Definiendo V_g como la velocidad del fuselaje respecto a tierra, V_a como la velocidad del fuselaje en relación con la masa de aire circundante, y V_w como la velocidad del viento. La relación entre estas velocidades viene dada por

$$V_g = V_a + V_w \quad (2.42)$$

Con el objetivo de poder realizar simulaciones, consideramos la velocidad del viento V_w como

$$W_g = V_{w_s} + V_{w_g} \quad (2.43)$$

donde V_{w_s} es un vector constante que representa un viento estable, y V_{w_g} un proceso estocástico que modela ráfagas de viento y otras perturbaciones atmosféricas. El viento estable se expresa típicamente en el sistema de referencia inercial como

$$V_{w_s}^i = \begin{bmatrix} w_{n_s} \\ w_{e_s} \\ w_{d_s} \end{bmatrix} \quad (2.44)$$

donde w_{n_s} es la velocidad del viento constante en la dirección norte, w_{e_s} es la velocidad del viento estable en la dirección este, y w_{d_s} es la velocidad del viento estable aguas abajo. La componente estocástica del viento se expresa en ejes cuerpo de la aeronave debido a que los efectos producidos son más frecuentes en la dirección longitudinal de la aeronave (aumento del ángulo de ataque inducido) que en sentido lateral o aguas abajo. Expresando en ejes cuerpo

$$V_{w_g}^b = \begin{bmatrix} u_{w_g} \\ v_{w_g} \\ w_{w_g} \end{bmatrix} \quad (2.45)$$

Los resultados experimentales indican que un buen modelo para esta componente no continua se obtiene haciendo pasar ruido blanco a través de un filtro lineal e invariante en el tiempo dado por el espectro de turbulencia de Von Karman. Desafortunadamente, dicho espectro no resulta en una función de transferencia racional. Una aproximación adecuada del modelo de von Karmen está dada por las funciones de transferencia de Dryden

$$H_u(s) = \sigma_u \sqrt{\frac{2V_a}{L_u}} \frac{1}{s + \frac{V_a}{L_u}} \quad (2.46)$$

$$H_v(s) = \sigma_v \sqrt{\frac{3V_a}{L_v}} \frac{\left(s + \frac{V_a}{\sqrt{3}L_v}\right)}{\left(s + \frac{V_a}{L_v}\right)^2} \quad (2.47)$$

$$H_w(s) = \sigma_w \sqrt{\frac{3V_a}{L_w}} \frac{\left(s + \frac{V_a}{\sqrt{3}L_w}\right)}{\left(s + \frac{V_a}{L_w}\right)^2} \quad (2.48)$$

donde σ_u , σ_v y σ_w son las intensidades de la turbulencia a lo largo de los ejes de la aeronave, L_u , L_v y L_w son longitudes de onda espaciales y V_a la velocidad de la aeronave. Los modelos Dryden se implementan típicamente asumiendo una velocidad nominal constante V_{a0} . Los parámetros para el modelo de ráfaga Dryden se definen en *MIL-F-8785C*. A continuación, se muestra una tabla extraída de [6] donde se presentan parámetros apropiados para las distintas condiciones de turbulencia deseadas.

Tabla 2.3 Parámetros de ráfaga del modelo Dryden [6].

Tipo de ráfaga	Altitud (m)	$L_u = L_v$ (m)	L_w (m)	$\sigma_u = \sigma_v$ (m/s)	σ_w (m/s)
altitud baja, turbulencia débil	50	200	50	1.06	0.7
altitud baja, turbulencia moderada	50	200	50	2.12	1.4
altitud media, turbulencia débil	600	533	533	1.5	1.5
altitud media, turbulencia moderada	600	533	533	3.0	3.0

Las componentes constantes del viento se giran desde el sistema de referencia inercial al sistema de ejes cuerpo y se agregan a las componentes de ráfaga para producir el viento total en ejes cuerpo. La combinación de términos estables y ráfagas se puede expresar matemáticamente como

$$V_w^b = \begin{bmatrix} u_w \\ v_w \\ w_w \end{bmatrix} = \mathbf{R}_v^b(\phi, \theta, \psi) \begin{bmatrix} w_{n_s} \\ w_{e_s} \\ w_{d_s} \end{bmatrix} + \begin{bmatrix} u_{g_s} \\ v_{g_s} \\ w_{w_s} \end{bmatrix} \quad (2.49)$$

A partir de las componentes de la velocidad V_w^b y la velocidad respecto a tierra V_g^b , se puede calcular las componentes del vector velocidad del aire como

$$V_a^b = \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} = \begin{bmatrix} u - u_w \\ v - v_w \\ w - w_w \end{bmatrix} \quad (2.50)$$

Partiendo de las componentes del vector velocidad del aire V_a^b se puede calcular el valor de la velocidad V_a , el ángulo de ataque α y ángulo de resbalamiento β

$$V_a = \sqrt{u_r^2 + v_r^2 + w_r^2} \quad (2.51)$$

$$\alpha = \tan^{-1}\left(\frac{w_r}{u_r}\right) \quad (2.52)$$

$$\beta = \sin^{-1}\left(\frac{v_r}{\sqrt{u_r^2 + v_r^2 + w_r^2}}\right) \quad (2.53)$$

Estas expresiones de V_a , α y β se emplean para calcular las fuerzas y momentos aerodinámicos sobre la aeronave.

2.3 Resumen

La fuerza y momento total que actúa sobre la aeronave se puede escribir como

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} -mg \sin(\theta) \\ mg \cos(\theta) \sin(\phi) \\ mg \cos(\theta) \cos(\phi) \end{bmatrix} + \frac{1}{2} \rho V_a^2 S \begin{bmatrix} C_X(\alpha) + C_{X_q}(\alpha) \frac{c}{2V_a} q + C_{X_{\delta_e}} \delta_e \\ C_{Y_0} + C_{Y_\beta} \beta + C_{Y_p} \frac{b}{2V_a} p + C_{Y_r} \frac{b}{2V_a} r + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r \\ C_Z(\alpha) + C_{Z_q}(\alpha) \frac{c}{2V_a} q + C_{Z_{\delta_e}} \delta_e \end{bmatrix} + \frac{1}{2} \rho S_{prop} C_{prop} \begin{bmatrix} (k_{motor} \delta_t)^2 - V_a^2 \\ 0 \\ 0 \end{bmatrix} \quad (2.54)$$

$$\begin{bmatrix} l \\ m \\ n \end{bmatrix} = \frac{1}{2} \rho V_a^2 S \begin{bmatrix} b \left(C_{l_0} + C_{l_\beta} \beta + C_{l_p} \frac{b}{2V_a} p + C_{l_r} \frac{b}{2V_a} r + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r \right) \\ c \left(C_{m_0} + C_{m_\alpha} \alpha + C_{m_q} \frac{c}{2V_a} q + C_{m_{\delta_e}} \delta_e \right) \\ b \left(C_{n_0} + C_{n_\beta} \beta + C_{n_p} \frac{b}{2V_a} p + C_{n_r} \frac{b}{2V_a} r + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r \right) \end{bmatrix} \quad (2.55)$$

donde

$$\begin{aligned} C_X(\alpha) &= -C_D(\alpha) \cos(\alpha) + C_L(\alpha) \sin(\alpha) \\ C_{X_q}(\alpha) &= -C_{D_q}(\alpha) \cos(\alpha) + C_{L_q}(\alpha) \sin(\alpha) \\ C_{X_{\delta_e}}(\alpha) &= -C_{D_{\delta_e}} \cos(\alpha) + C_{L_{\delta_e}} \sin(\alpha) \\ C_Z(\alpha) &= -C_D(\alpha) \sin(\alpha) + C_L(\alpha) \cos(\alpha) \\ C_{Z_q}(\alpha) &= -C_{D_q} \sin(\alpha) - C_{L_q} \cos(\alpha) \\ C_{Z_{\delta_e}}(\alpha) &= -C_{D_{\delta_e}} \sin(\alpha) - C_{L_{\delta_e}} \cos(\alpha) \end{aligned} \quad (2.56)$$

Notas y referencias

Las ecuaciones desarrolladas en este capítulo han sido extraídas de [2]

2.4 Trimado UAV

En este apartado vamos a calcular las condiciones de trimado para el UAV. Este cálculo es indispensable para la posterior implementación de controladores en el UAV, ya que es necesario obtener las variables de referencia necesarias para poder realizar el control en bucle cerrado, tal y como se desarrolla en el Capítulo 3.

Vamos a suponer que el UAV va a realizar un vuelo a 100m de altitud (por simplificación consideraremos que el territorio a sobrevolar se va a encontrar a nivel del mar). Las ecuaciones necesarias para calcular el valor de las variables de control δ_t , δ_a , δ_e van a ser una simplificación de las ecuaciones desarrolladas durante el Capítulo, ya que permiten un cálculo muy sencillo y a la vez muy aproximado sobre el estado de trimado del UAV.

Las ecuaciones que nos permiten calcular dicho punto son las correspondientes a vuelo de crucero y para velocidad de 15 m/s, a partir de las cuales se obtienen los siguientes valores:

$$\begin{aligned} \delta_t^* &= 0.34875 \\ \delta_a^* &= -4.1359 \cdot 10^{-25} \text{ rad} \sim 0 \text{ rad} \\ \delta_e^* &= -0.0072586 \text{ rad} \end{aligned}$$

3 Control y Guiado: Diseño de Autopiloto

En términos generales, un piloto automático es un sistema utilizado para guiar una aeronave sin la ayuda de un piloto. Para los UAV, el piloto automático tiene el control completo de la aeronave durante todas las fases del vuelo. Si bien algunas funciones de control pueden residir en la estación de control de tierra, la parte del piloto automático del sistema de control de UAV reside a bordo del UAV.

En este capítulo se diseñará un autopiloto para UAVs basado en control mediante bucle cerrado. El objetivo de este diseño es implementar una ley de control que se encargue de calcular en todo momento el valor de las variables de control del sistema ($\delta_t, \delta_a, \delta_e$) para que la aeronave vuele de manera autónoma. El hecho de que la variable de control δ_r no se haya mencionado es debido a que el UAV estudiado se trata de un ala volante, y por tanto, carece de timón de profundidad. Para la implementación de los controladores se va a hacer uso del Capítulo 6 de [2].

3.1 Autopiloto Longitudinal

El objetivo en el diseño del piloto automático longitudinal será regular la velocidad del aire y la altitud utilizando la señal PWM (δ_t) y elevadores (δ_e) como actuadores. El método utilizado para regular la altitud y la velocidad del aire depende del error de altitud. Los regímenes de vuelo se muestran en la figura 3.1

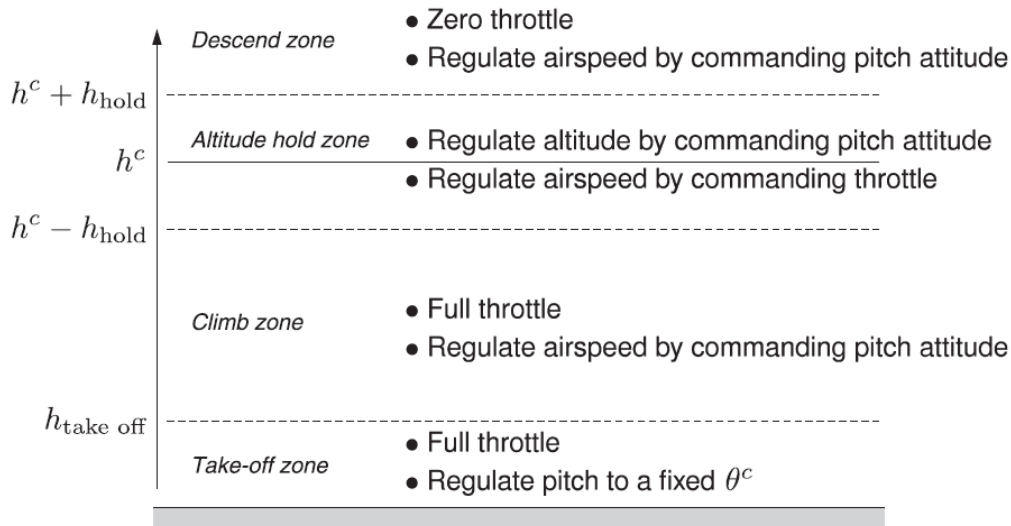


Figura 3.1 Regímenes de vuelo para el autopiloto longitudinal..

En la zona de despegue, se ordena empuje máximo ($\delta_t = 1$) y la actitud de cabeceo se regula a un ángulo de cabeceo fijo θ_c utilizando los elevadores. El objetivo en la zona de ascenso es maximizar la velocidad

de ascenso dadas las condiciones atmosféricas actuales. Para maximizar la velocidad de ascenso, se ordena empuje máximo y la velocidad del aire se regula utilizando el ángulo de cabeceo.

La zona de descenso es similar a la zona de ascenso, excepto que la señal PWM se ordena a cero. En la zona de retención de altitud, la velocidad del aire se regula ajustando la señal PWM, y la altitud se regula mediante el ángulo de cabeceo.

3.1.1 Controlador de Cabeceo

A partir de la Fig. (3.2) se puede obtener la función de transferencia que permite pasar de θ^c a θ

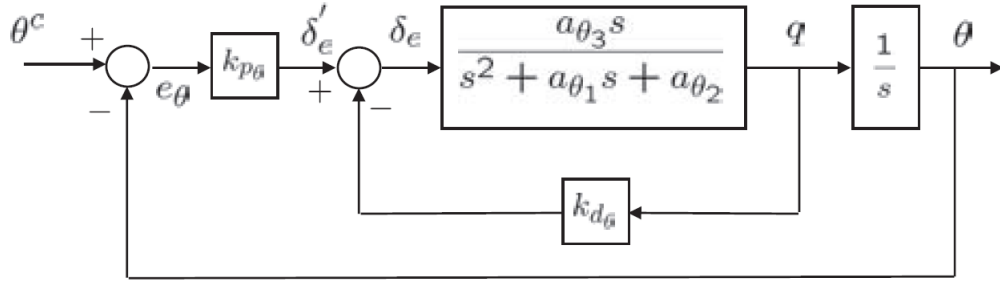


Figura 3.2 Diagrama de bloques del controlador de cabeceo..

$$H_{\theta/\theta^c}(s) = \frac{k_{p_\theta} a_{\theta_3}}{s^2 + (a_{\theta_1} + k_{d_\theta} a_{\theta_3})s + (a_{\theta_2} + k_{p_\theta} a_{\theta_3})} \quad (3.1)$$

Si queremos expresar la función de transferencia en la forma canónica

$$H(s) = \frac{K_{\theta_{DC}} \omega_{n_\theta}^2}{s^2 + 2\zeta_\theta \omega_{n_\theta} s + \omega_{n_\theta}^2} \quad (3.2)$$

Igualando términos se obtiene

$$\omega_{n_\theta}^2 = a_{\theta_2} + k_{p_\theta} a_{\theta_3} \quad (3.3a)$$

$$2\zeta_\theta \omega_{n_\theta} = a_{\theta_1} + k_{d_\theta} a_{\theta_3} \quad (3.3b)$$

Si configuramos la ganancia proporcional para evitar la saturación cuando se experimenta el error máximo de entrada, obtenemos

$$k_{p_\theta} = \frac{\delta_e^{max}}{e_\theta^{max}} \text{sign}(a_{\theta_3}) \quad (3.4)$$

donde se toma el signo de a_{θ_3} , ya que a_{θ_3} se basa en $C_{m_{\delta_e}}$ que suele ser negativo. Para garantizar la estabilidad, k_{p_θ} y a_{θ_3} deben ser del mismo signo. A partir de la ecuación (3.3a) se puede calcular el límite del ancho de banda del bucle de cabeceo como

$$\omega_{n_\theta} = \sqrt{a_{\theta_2} + \frac{\delta_e^{max}}{e_\theta^{max}} |a_{\theta_3}|} \quad (3.5)$$

y resolviendo la ecuación (3.3b)

$$k_{d_\theta} = \frac{2\zeta_\theta \omega_{n_\theta} - a_{\theta_1}}{a_{\theta_3}} \quad (3.6)$$

En resumen, conociendo el límite de saturación del actuador δ_e^{max} y el máximo error de cabeceo, se puede seleccionar e_θ^{max} para determinar la ganancia proporcional k_{p_θ} y el ancho de banda del bucle de cabeceo. Seleccionando la relación de amortiguamiento deseada ζ_θ se fija el valor de ganancia derivativa k_{d_θ} .

La ganancia DC viene dada por

$$K_{\theta DC} = \frac{k_{p\theta} a_{\theta 3}}{(a_{\theta 2} + k_{p\theta} a_{\theta 3})} \quad (3.7)$$

Por tanto, se realiza el control mediante bucle cerrado con un controlador PD, siendo la respuesta de los elevadores

$$\delta_e = k_{p\theta}(\theta^c - \theta) - k_{d\theta} \dot{\theta} \quad (3.8)$$

No se ha empleado el término integral debido a que puede limitar considerablemente el ancho de banda del bucle interno.

3.1.2 Controlador de Altitud mediante Cabeceo

El controlador de altitud se basa en un controlador PI en el que a partir de la entrada de altitud de referencia h^c , se obtiene el ángulo de cabeceo de referencia θ^c entrante al controlador de cabeceo. El diseño del controlador de altitud se encuentra en la Fig. (3.3). De forma análoga a como se ha realizado en el controlador de cabeceo,

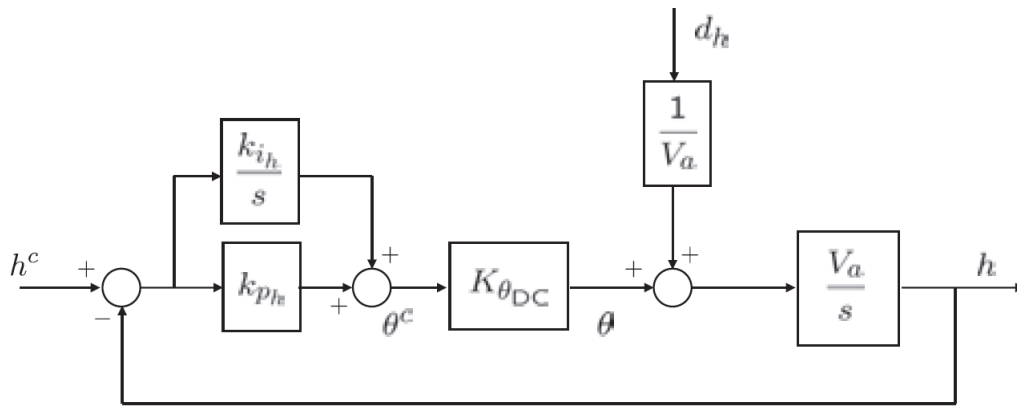


Figura 3.3 Diagrama de bloques del controlador de altitud..

obtenemos las siguientes expresiones

$$h(s) = \left(\frac{K_{\theta DC} V_a k_{p_h} \left(s + \frac{k_{i_h}}{k_{p_h}} \right)}{s^2 + K_{\theta DC} V_a k_{p_h} s + K_{\theta DC} V_a k_{i_h}} \right) h^c(s) + \left(\frac{s}{s^2 + K_{\theta DC} V_a k_{p_h} s + K_{\theta DC} V_a k_{i_h}} \right) d_h(s) \quad (3.9)$$

Las ganancias k_{p_h} y k_{i_h} deben elegirse de modo que el ancho de banda del bucle de altitud-cabeceo sea menor que el ancho de banda del bucle de actitud-cabeceo. Calculamos la frecuencia natural a partir de la separación del ancho de banda W_h

$$\omega_{n_h} = \frac{\omega_{n_\theta}}{W_h} \quad (3.10)$$

Volviendo a hacer uso de la ecuación (3.2) calculamos los coeficientes

$$\omega_{n_h}^2 = K_{\theta DC} V_a k_{i_h} \quad (3.11a)$$

$$2\zeta_h \omega_{n_h} = K_{\theta DC} V_a k_{p_h} \quad (3.11b)$$

y despejamos el valor de las ganancias k_{p_h} y k_{i_h}

$$k_{i_h} = \frac{\omega_{n_h}^2}{K_{\theta DC} V_a} \quad (3.12)$$

$$k_{ph} = \frac{2\zeta_h \omega_{nh}}{K_{\theta DC} V_a} \quad (3.13)$$

Por lo tanto, al seleccionar la relación de amortiguamiento deseada ζ_h y la separación del ancho de banda W_h se fija el valor para k_{ph} y k_{ih} .

La salida del bucle cerrado de altitud-cabeceo es

$$\theta^c = k_{ph}(h^c - h) + \frac{k_{ih}}{s}(h^c - h) \quad (3.14)$$

3.1.3 Controlador de Velocidad mediante Cabeceo

El controlador de velocidad mediante el ángulo de cabeceo se basa en un controlador PI en el que a partir de la entrada de velocidad de referencia \bar{V}_a^c , se obtiene el ángulo de cabeceo de referencia θ^c entrante al controlador de cabeceo. El diseño de dicho controlador se encuentra en la Fig. (3.4).

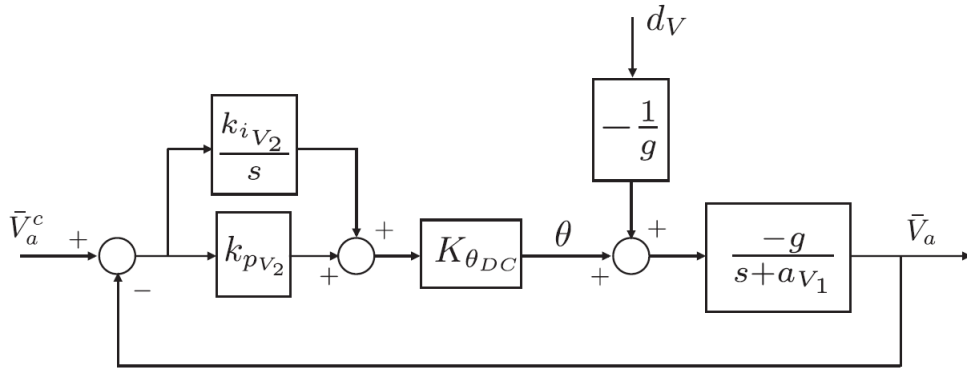


Figura 3.4 Diagrama de bloques del controlador de velocidad mediante cabeceo..

En el dominio de Laplace

$$\begin{aligned} \bar{V}_a(s) = & \left(\frac{-K_{\theta DC} g k_{pV_2} \left(s + \frac{k_{iV_2}}{k_{pV_2}} \right)}{s^2 + (a_{V_1} - K_{\theta DC} g k_{pV_2})s - K_{\theta DC} g k_{pV_2}} \right) \bar{V}_a^c(s) \\ & + \left(\frac{s}{s^2 + (a_{V_1} - K_{\theta DC} g k_{pV_2})s - K_{\theta DC} g k_{pV_2}} \right) d_V(s) \end{aligned} \quad (3.15)$$

Las ganancias k_{pV_2} y k_{iV_2} deben elegirse de modo que el ancho de banda del bucle de altitud-cabeceo sea menor que el ancho de banda del bucle de actitud-cabeceo. Calculamos la frecuencia natural a partir de la separación del ancho de banda W_{V_2}

$$\omega_{nV_2} = \frac{\omega_{n\theta}}{W_{V_2}} \quad (3.16)$$

Se vuelven a calcular las ganancias

$$k_{iV_2} = -\frac{\omega_{nV_2}^2}{K_{\theta DC} g} \quad (3.17)$$

$$k_{pV_2} = \frac{a_{V_1} - 2\zeta_{V_2} \omega_{nV_2}}{K_{\theta DC} g} \quad (3.18)$$

Seleccionando la relación de amortiguamiento deseada ζ_{V_2} y la separación del ancho de banda W_{V_2} se fija el valor para $k_{p_{V_2}}$ y $k_{i_{V_2}}$. La salida del bucle cerrado velocidad-cabeceo es

$$\theta^c = k_{p_{V_2}}(V_a^c - V_a) + \frac{k_{i_{V_2}}}{s}(V_a^c - V_a) \quad (3.19)$$

3.1.4 Controlador de Velocidad mediante Empuje

El controlador de velocidad mediante empuje se basa en un controlador PI en el que a partir de la entrada de velocidad de referencia \bar{V}_a^c , se obtiene el valor de la señal PWM δ_t . El diseño de dicho controlador se encuentra en la Fig. (3.5). En el dominio de Laplace

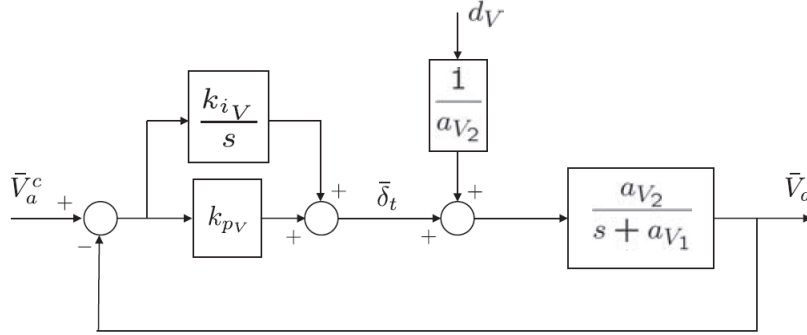


Figura 3.5 Diagrama de bloques del controlador de velocidad mediante empuje..

$$\begin{aligned} \bar{V}_a(s) = & \left(\frac{a_{V_2}(k_{p_V}s + k_{i_V})}{s^2 + (a_{V_1} + a_{V_2}k_{p_V})s + a_{V_2}k_{i_V}} \right) \bar{V}_a^c(s) \\ & + \left(\frac{1}{s^2 + (a_{V_1} + a_{V_2}k_{p_V})s + a_{V_2}k_{i_V}} \right) d_V(s) \end{aligned} \quad (3.20)$$

Si a_{V_1} y a_{V_2} son conocidas, las ganancias k_{p_V} y k_{i_V} se obtienen mediante las mismas técnicas comentadas anteriormente

$$k_{i_V} = \frac{\omega_{n_V}^2}{a_{V_2}} \quad (3.21)$$

$$k_{p_V} = \frac{2\zeta_V\omega_{n_V} - a_{V_1}}{a_{V_2}} \quad (3.22)$$

Los parámetros de diseño para este bucle cerrado son la relación de amortiguamiento ζ_V y frecuencia natural ω_{n_V} . La salida del bucle cerrado velocidad-empuje es

$$\delta_t = \delta_t^* + k_{p_V}(V_a^c - V_a) + \frac{k_{i_V}}{s}(V_a^c - V_a) \quad (3.23)$$

donde δ_t^* es el valor de la señal PWM en condición de trimado del UAV.

3.1.5 Máquina de Estados de Control de Altitud

En este apartado se va a desarrollar el bloque correspondiente a la máquina de estados de control longitudinal. Se trata de un bloque necesario para poder implementar en el autopiloto los regímenes de la Fig. (3.1). En función de la condición de vuelo que se encuentre el UAV, la máquina de estados se encarga de indicar qué controladores deben actuar y cuales no.

Podemos ver una representación de dicha máquina de estados en la Fig. (3.6). Se observa que cada régimen de vuelo se encuentra caracterizado por un ley de control:

- **Zona de descenso:** empuje nulo y control de velocidad mediante cabeceo.
- **Zona de retención de altitud:** control de velocidad mediante empuje y control de altitud mediante cabeceo.
- **Zona de ascenso:** empuje máximo y control de velocidad mediante cabeceo.
- **Zona de aterrizaje:** empuje máximo y ángulo de cabeceo fijo.

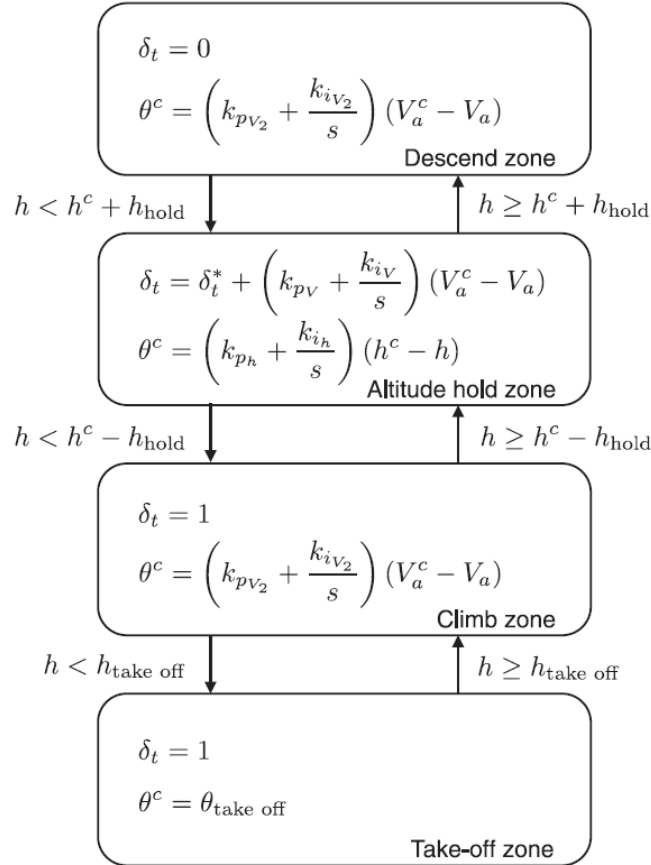


Figura 3.6 Máquina de estados para control longitudinal..

3.2 Autopiloto Lateral

El objetivo del autopiloto lateral es la de obtener una ley de control que se encargue de la deflexión automática de las superficies de control lateral. Al tratarse de un ala volante, sólo tendremos control mediante los alerones δ_a . Por esta razón, es necesario implementar una ley de control en cascada que permita obtener la deflexión de los alerones a partir de un ángulo de guiñada de referencia. Para ello es necesario diseñar un controlador de alabeo cuya entrada de referencia es la salida del controlador de guiñada, es decir, el control del ángulo de guiñada y alabeo se encuentra acoplado.

3.2.1 Controlador de Alabeo

El controlador de alabeo se basa en un controlador PID en el que a partir de la entrada de alabeo de referencia ϕ^c , se obtiene la deflexión de los alerones δ_a necesaria. El diseño de dicho controlador se encuentra en la Fig. (3.7).

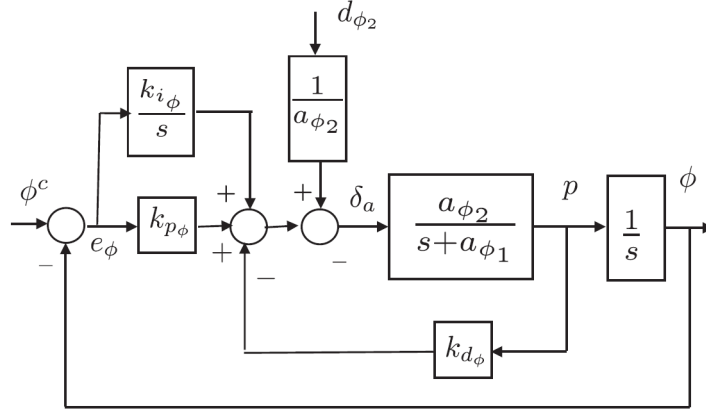


Figura 3.7 Diagrama de bloques del controlador de alabeo..

En el dominio de Laplace

$$\phi = \left(\frac{s}{s^3 + (a_{\phi_1} + a_{\phi_2} k_{d_\phi})s^2 + a_{\phi_2} k_{p_\phi} s + a_{\phi_2} k_{i_\phi}} \right) d_{\phi_2} + \left(\frac{a_{\phi_2} k_{p_\phi} \left(s + \frac{k_{i_\phi}}{k_{p_\phi}} \right)}{s^3 + (a_{\phi_1} + a_{\phi_2} k_{d_\phi})s^2 + a_{\phi_2} k_{p_\phi} s + a_{\phi_2} k_{i_\phi}} \right) \phi^c \quad (3.24)$$

De forma análoga a como se hizo en el controlador de cabeceo, obtenemos los parámetros que definen el controlador

$$k_{p_\phi} = \frac{\delta_a^{max}}{e_\phi^{max}} \text{sign}(a_{\phi_2}) \quad (3.25)$$

$$\omega_{n_\phi} = \sqrt{\frac{\delta_a^{max}}{e_\phi^{max}} |a_{\phi_2}|} \quad (3.26)$$

$$k_{d_\phi} = \frac{2\zeta_\phi \omega_{n_\phi} - a_{\phi_1}}{a_{\phi_2}} \quad (3.27)$$

donde la relación de amortiguamiento ζ_ϕ es un parámetro de diseño.

En cuanto al término integral del controlador, éste puede ser seleccionado empleando el método del lugar de las raíces [2].

La salida del bucle cerrado de alabeo es

$$\delta_a = k_{p_\phi}(\phi^c - \phi) + \frac{k_{i_\phi}}{s}(\phi^c - \phi) - k_{d_\phi} p \quad (3.28)$$

3.2.2 Controlador de Rumbo

Este controlador tiene como función calcular la entrada necesaria al controlador de alabeo ϕ^c para obtener el rumbo deseado debido a que el UAV no dispone de timón de profundidad. Se va emplear un controlador PI cuya entrada es el rumbo de referencia χ^c , cuyo modelo de diagrama de bloques se puede ver en la Fig. (3.8).

Volviendo a elegir el valor de la frecuencia natural ω_{n_χ} y la relación de amortiguamiento ζ_χ podemos calcular el valor de las ganancias k_{p_χ} y k_{i_χ} . Calculamos el valor de la frecuencia natural a partir de la separación del ancho de banda W_χ .

$$\omega_{n_\chi} = \frac{\omega_{n_\phi}}{W_\chi} \quad (3.29)$$

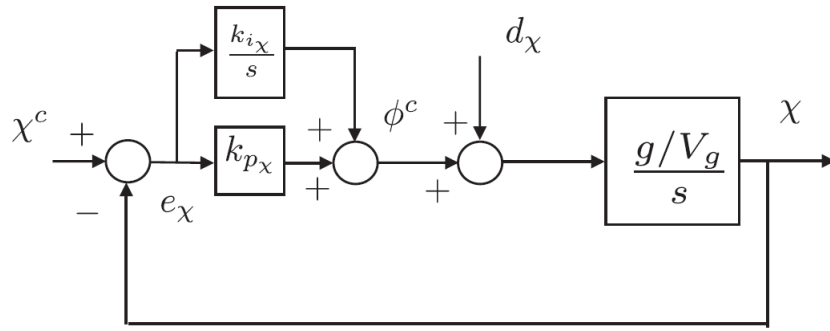


Figura 3.8 Diagrama de bloques del controlador de rumbo..

$$k_{p_\chi} = 2\zeta_\chi \omega_{n_\chi} \frac{V_g}{g} \quad (3.30)$$

$$k_{i_\chi} = \omega_{n_\chi}^2 \frac{V_g}{g} \quad (3.31)$$

La salida del bucle cerrado de rumbo es

$$\phi^c = k_{p_\chi}(\chi^c - \chi) + \frac{k_{i_\chi}}{s}(\chi^c - \chi) \quad (3.32)$$

4 Guiado: Diseño del Sistema de Misión

El objetivo de este proyecto es la de una vez modelado nuestro UAV, que éste realice una determinada misión en el plano horizontal: se le da como entrada una serie de puntos de paso (*waypoints*) y el UAV debe ser capaz de ir de uno a otro. Para ello, es necesario instalar un sistema de misión que sea capaz de obtener las entradas de referencia necesarias para que el autopiloto pueda actuar sobre las variables de control. En concreto, el sistema de misión se encarga de calcular la altitud y rumbo de referencia para cada instante de simulación a partir de las variables de estado del UAV. Hay que destacar que el objetivo de este proyecto es que el UAV pase por los waypoints establecidos sin tener en cuenta el tiempo de vuelo, siendo esta una posible futura mejora del sistema.

En la literatura podemos encontrar diversos algoritmos para el diseño del sistema de misión en función del nivel de complejidad que queramos implementar. Por ejemplo, en este proyecto se ha optado por un sistema de guiado mediante segmentos rectilíneos debido a su mayor facilidad. Otros métodos pueden estar basados en combinación de trayectorias rectilíneas y circulares o técnicas de detección de obstáculos [2].

4.1 Algoritmo para cálculo de ruta

Este algoritmo se encarga de calcular en cada instante la altitud y rumbo de referencia para ir hacia el waypoint objetivo mediante trayectorias rectilíneas. Un camino en línea recta se describe mediante dos vectores, pertenecientes a \mathbb{R}^3 , de forma

$$P_{line}(\mathbf{r}, \mathbf{q}) = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = \mathbf{r} + \lambda \mathbf{q}, \lambda \in \mathbb{R}\}$$

donde $\mathbf{r} \in \mathbb{R}^3$ es el origen del camino, y $\mathbf{q} \in \mathbb{R}^3$ es la dirección deseada de vuelo. El rumbo de la línea $P_{line}(\mathbf{r}, \mathbf{q})$, medido desde el norte, viene dado por

$$\chi_q = \text{atan2}\left(\frac{q_e}{q_n}\right) \quad (4.1)$$

donde $\mathbf{q} = (q_n, q_e, q_d)^T$ expresa las coordenadas en ejes NED del vector de dirección. En la Fig. (4.1) podemos ver una representación de la geometría empleada para el cálculo de la ruta a implementar.

Aplicando geometría obtenemos la expresión de la altitud de referencia en cada instante

$$h^c(\mathbf{r}, \mathbf{p}, \mathbf{q}) = -r_d + \sqrt{s_n^2 + s_e^2} \left(\frac{|q_d|}{\sqrt{q_n^2 + q_e^2}} \right) \quad (4.2)$$

donde $\mathbf{p} = (p_n, p_e, p_d)^T$ es la posición del UAV en ejes NED.

En cuanto al rumbo necesario para llegar al waypoint objetivo hay que hacer nulo el error cometido en la componente horizontal del plano inercial norte-este (e_{py}). De acuerdo a [?] la dinámica del sistema viene dada por

$$\dot{e}_{py} = V_g \sin(\chi - \chi_q) \quad (4.3a)$$

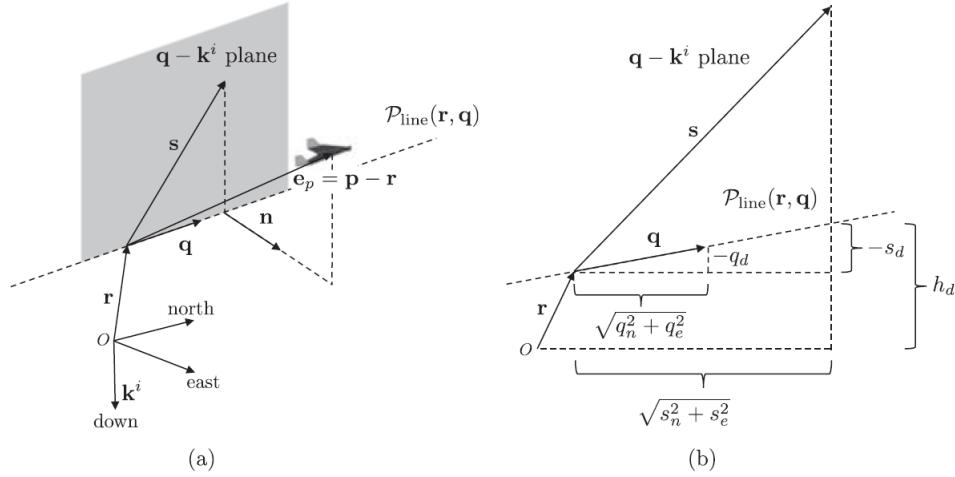


Figura 4.1 Altitud deseada para diseño de ruta longitudinal..

$$\ddot{\chi} = b_{\dot{\chi}}(\dot{\chi}^c - \dot{\chi}) + b_{\chi}(\chi^c - \chi) \quad (4.3b)$$

Expresando el rumbo deseado para el UAV en función de e_{py}

$$\chi_d(e_{py}) = -\chi^\infty \frac{2}{\pi} \tan^{-1}(k_{path} e_{py}) \quad (4.4)$$

donde k_{path} es una constante positiva que define la transición desde χ^∞ a cero. Valores pequeños definen transiciones suaves, mientras que valores mayores provocan transiciones abruptas. Debido a que $\chi^\infty \in (0, \pi/2]$, entonces para todos los valores de e_{py} se cumple

$$-\frac{\pi}{2} < \chi^\infty \frac{2}{\pi} \tan^{-1}(k_{path} e_{py}) < \frac{\pi}{2}$$

Aplicando la función de Lyapunov $W(e_{py}) = \frac{1}{2} e_{py}^2$ para demostrar que si $\chi = \chi_q + \chi^d(e_{py})$, entonces $e_{py} \rightarrow 0$ asintóticamente, por lo tanto

$$\dot{W} = -V_g e_{py} \sin\left(\chi^\infty \frac{2}{\pi} \tan^{-1}(k_{path} e_{py})\right) \quad (4.5)$$

es menor que cero para $e_{py} \neq 0$. El rumbo deseado para el seguimiento de la ruta viene dado por

$$\chi^c(t) = \chi_q - \chi^\infty \frac{2}{\pi} \tan^{-1}(k_{path} e_{py}(t)) \quad (4.6)$$

Si χ_q es calculado a partir de la ecuación (4.1), la función $atan2$ puede devolver un ángulo comprendido entre $\pm\pi$ que puede dar lugar a problemas en la ruta del UAV. Para solucionar dicho problema, χ_q es calculado como

$$\chi_q = atan2\left(\frac{q_e}{q_n}\right) + 2\pi m \quad (4.7)$$

donde $m \in \mathbb{N}$ se elige de tal forma que $-\pi \leq \chi_q - \chi \leq \pi$. Para un desarrollo con mayor detalle del algoritmo empleado se recomienda consultar [2].

En la Fig. (4.2) se muestra el algoritmo empleado de forma resumida:

Algorithm 3 Straight-line Following: $[h^c, \chi^c] = \text{followStraightLine}(\mathbf{r}, \mathbf{q}, \mathbf{p}, \chi)$

Input: Path definition $\mathbf{r} = (r_n, r_e, r_d)^\top$ and $\mathbf{q} = (q_n, q_e, q_d)^\top$, MAV position $\mathbf{p} = (p_n, p_e, p_d)^\top$, course χ , gains $\chi_\infty, k_{\text{path}}$, sample rate T_s .

- 1: Compute commanded altitude using equation (10.5).
- 2: $\chi_q \leftarrow \text{atan2}(q_e, q_n)$
- 3: **while** $\chi_q - \chi < -\pi$ **do**
- 4: $\chi_q \leftarrow \chi_q + 2\pi$
- 5: **end while**
- 6: **while** $\chi_q - \chi > \pi$ **do**
- 7: $\chi_q \leftarrow \chi_q - 2\pi$
- 8: **end while**
- 9: $e_{py} \leftarrow -\sin \chi_q (p_n - r_n) + \cos \chi_q (p_e - r_e)$
- 10: Compute commanded course angle using equation (10.8).
- 11: **return** h^c, χ^c

Figura 4.2 Algoritmo empleado para el cálculo de la ruta en línea recta..

Una vez definidos la altitud y rumbos de referencias necesarios para llegar al waypoint objetivo, es necesario definir cómo el sistema de misión es capaz de comprobar si el UAV ha llegado o no al waypoint de destino. Para ello se ha decidido hacer uso de un algoritmo bastante sencillo pero que ha proporcionado buenos resultados en las distintas simulaciones realizadas.

El criterio de este algoritmo para actualizar la senda de vuelo y permitir a la aeronave continuar la ruta se basa en discernir si esta ha traspasado o no la línea imaginaria que separa el semiplano que contiene el waypoint al cual se dirige y el anterior, y el semiplano que contiene el waypoint al cual se dirige y el siguiente. En resumen, si ha pasado de un semiplano a otro.

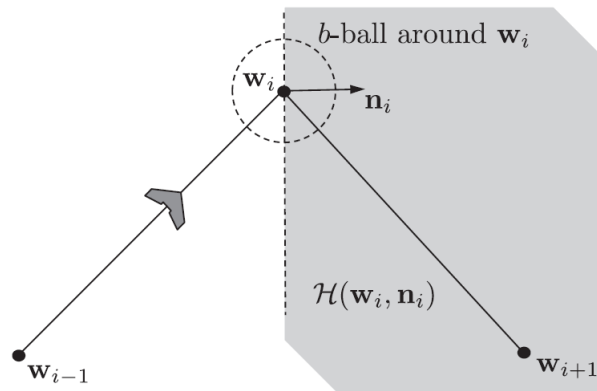


Figura 4.3 Criterio de actualización de segmento de ruta...

En la Fig. (4.4) se muestra el algoritmo implementado.

Algorithm 5 Follow Waypoints: $(\mathbf{r}, \mathbf{q}) = \text{followWpp}(\mathcal{W}, \mathbf{p})$ **Input:** Waypoint path $\mathcal{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_N\}$, MAV position

$$\mathbf{p} = (p_n, p_e, p_d)^\top.$$

Require: $N \geq 3$

```

1: if New waypoint path  $\mathcal{W}$  is received then
2:   Initialize waypoint index:  $i \leftarrow 2$ 
3: end if
4:  $\mathbf{r} \leftarrow \mathbf{w}_{i-1}$ 
5:  $\mathbf{q}_{i-1} \leftarrow \frac{\mathbf{w}_i - \mathbf{w}_{i-1}}{\|\mathbf{w}_i - \mathbf{w}_{i-1}\|}$ 
6:  $\mathbf{q}_i \leftarrow \frac{\mathbf{w}_{i+1} - \mathbf{w}_i}{\|\mathbf{w}_{i+1} - \mathbf{w}_i\|}$ 
7:  $\mathbf{n}_i \leftarrow \frac{\mathbf{q}_{i-1} + \mathbf{q}_i}{\|\mathbf{q}_{i-1} + \mathbf{q}_i\|}$ 
8: if  $\mathbf{p} \in \mathcal{H}(\mathbf{w}_i, \mathbf{n}_i)$  then
9:   Increment  $i \leftarrow (i + 1)$  until  $i = N - 1$ 
10: end if
11: return  $\mathbf{r}, \mathbf{q} = \mathbf{q}_{i-1}$  at each time step

```

Figura 4.4 Algoritmo de gestión de ruta..

4.2 Implementación de un Sistema Planificador de Vuelo TSP

Una vez explicado el algoritmo empleado para obtener las variables de referencia para controlar el vuelo del UAV, ahora se va a desarrollar el método empleado para el cálculo de la ruta óptima que pasa por cada uno de los puntos de paso definidos, es decir, un sistema planificador de vuelo para minimizar los costes de vuelo asociados a una determinada misión, recogido en [7]. Este problema matemático ha sido de un gran interés durante la segunda mitad del Siglo XX y se le conoce como TSP (*Travelling Salesman Problem*), que responde a la siguiente pregunta: dada una lista de ciudades y las distancias entre cada par de ellas, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y al finalizar regresa a la ciudad origen? Este es un problema NP-Hard dentro en la optimización combinatoria, muy importante en la investigación de operaciones y en la ciencia de la computación.

El objetivo es encontrar un recorrido completo que conecte todos los nodos de una red, visitándolos tan solo una vez y volviendo al punto de partida, y que además minimice la distancia total de la ruta, o el tiempo total del recorrido. Este tipo de problemas tiene gran aplicación en el ámbito de la logística y distribución, así como en la programación de curvas de producción. De esta forma, vamos a plantear un mapa del terreno con

una serie de waypoints que corresponden con zonas o áreas significativas que hay que visitar. Este algoritmo se basa en elaborar un plan de vuelo en el que la aeronave visite todas las áreas una sola vez, y que, además, la trayectoria realizada sea óptima en términos de distancia y tiempo. Como ejemplo práctico de aplicación de este algoritmo, podemos considerar misiones de rescate, misiones de exploración, misiones de agricultura o misiones militares.

Al tratarse de un problema NP-Hard, el número de posibles soluciones crece exponencialmente con el número de nodos del grafo (waypoints) y rápidamente sobrepasa las capacidades de cálculo de los ordenadores más potentes. No existen soluciones exactas, sin embargo, existen Heurísticos que se aproximan bastante a la solución óptima. El único método que nos da la solución exacta es el denominado de fuerza bruta, que estudia todos los caminos posibles y se queda con el de menor recorrido. Evidentemente, este método no es factible computacionalmente, ya que, cuando el número de puntos es considerable, no existe computadora capaz de estudiar todos los caminos posibles. Por otro lado, existen aproximaciones como la del vecino más

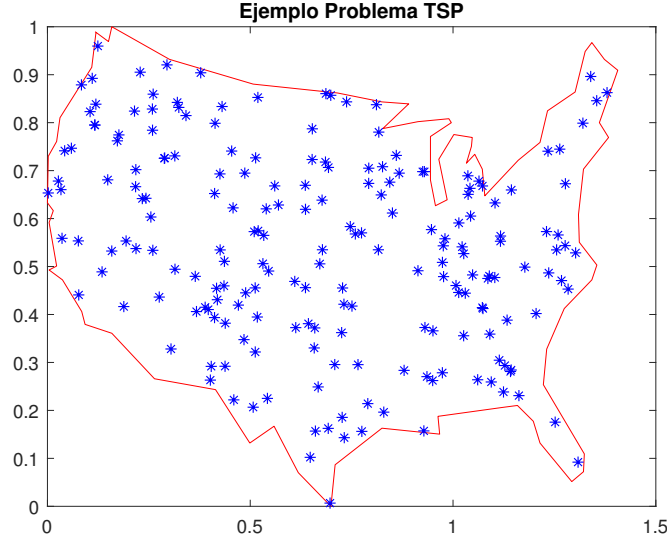


Figura 4.5 *Ejemplo del Problema TSP..*

cercano, que estudia en cada waypoint el punto más cercano. Sin embargo, el mínimo local no garantiza el mínimo global, por lo que esta solución no converge al óptimo.

El método empleado para obtener una solución aproximada es un método de programación lineal con la restricción de que la aeronave visite todos los waypoints en una única ocasión. Además se considera que la distancia recorrida para ir de un nodo i a un nodo j es exactamente la misma que la recorrida para ir de un nodo j a un nodo i . Matemáticamente el problema se define de la siguiente forma:

$$\min \sum_{i=0}^n \sum_{j \neq i, j=0}^n C_{ij} X_{ij} \quad (4.8)$$

sujeto a

$$\sum_{i=0, i \neq ij}^n X_{ij} = 1 \quad (4.9a)$$

$$\sum_{j=0, j \neq i}^n X_{ij} = 1 \quad (4.9b)$$

$$u_i - u_j + nX_{ij} \leq n - 1 \quad ; \quad 1 \leq i \neq j \leq n \quad (4.9c)$$

donde C_{ij} es la distancia entre el punto i y el punto j , $X_{ij} = 1$ si se visita el waypoint j después de visitar el i , o $X_{ij} = 0$ en el caso de que no. Las dos primeras igualdades aseguran que desde cada waypoint $1, \dots, n$ se salga exactamente hacia un waypoint (ambas restricciones también implican que exista exactamente una salida desde el waypoint 0.) La última restricción obliga a que un solo camino cubra todos los waypoints y no dos o más caminos disjuntos cubran conjuntamente todos los waypoints.

A continuación se muestra un esquema de cuáles son los pasos a seguir para su resolución:

- Generar todos los viajes posibles, es decir, todos los pares distintos de paradas.
- Calcular la distancia para cada viaje.
- La función de coste para minimizar es la suma de las distancias de viaje para cada viaje en el Tour.
- Las variables de decisión son binarias, y se asocian con cada viaje, donde cada 1 representa un viaje que existe en el Tour, y cada 0 representa un viaje que no está en el Tour.
- Para asegurar que el tour incluye cada parada, hay que incluir la restricción lineal que cada parada está en exactamente dos viajes. Esto significa una llegada y una salida de la parada.

En la Fig. (4.6) podemos observar un ejemplo del algoritmo programado.

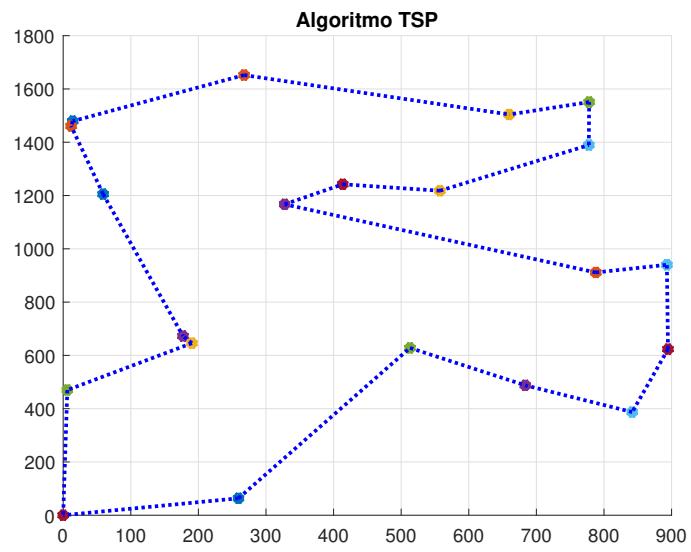


Figura 4.6 *Algoritmo TSP..*

5 Implementación de Sensores y Filtro de Kalman Extendido

Para llevar a cabo una simulación de mayor interés se va implementar en el modelo un observador basado en el Filtro de Kalman. Para ello es necesario dotar al UAV de una serie de sensores, los cuales se encargan de obtener las medidas necesarias para poder modelar el observador. Al tratarse de un sistema no lineal discreto, se hace uso del Filtro de Kalman Extendido, el cual se explicará en mayor detalle más adelante.

5.1 Sensores a bordo

Los sensores introducidos en el UAV corresponden a una serie de 5 sensores básicos de los cuales dispone cualquier aeronave: acelerómetro, giróscopo, altímetro, anemómetro y magnetómetro. Estos sensores proporcionan información necesaria en términos de actitud y sistemas de control de navegación. Se encuentran caracterizados por el tiempo de muestreo (T_s), que se ha considerado de 0.2 segundos en cada uno de ellos, y los parámetros de ruido que introducen en la medición. Los parámetros utilizados para modelarlos han sido extraídos del Apéndice H de [2].

5.1.1 Acelerómetro

Este sensor se encarga de medir las aceleraciones que experimenta la aeronave. Uno de los modelos más empleados es el acelerómetro piezoeléctrico, el cual se encarga de convertir el desplazamiento de una masa de prueba en una señal eléctrica. En aeronaves se suelen emplear tres acelerómetros colocados cerca del centro de masa y alineados con los ejes cuerpo de la aeronave.

El acelerómetro mide la aceleración del UAV en m/s^2 , siendo el modelo matemático empleado el siguiente

$$y_{accel,x} = \dot{u} + qw - rv + g \sin \theta + \eta_{accel,x} \quad (5.1a)$$

$$y_{accel,y} = \dot{v} + ru - pw - g \cos \theta \sin \phi + \eta_{accel,y} \quad (5.1b)$$

$$y_{accel,z} = \dot{w} - pv - qu - g \cos \theta \cos \phi + \eta_{accel,z} \quad (5.1c)$$

donde el ruido presente en los sensores es modelado mediante los parámetros $\eta_{accel,x}$, $\eta_{accel,y}$ y $\eta_{accel,z}$, que corresponden a ruido gaussiano con covarianzas $\sigma_{accel,x}^2$, $\sigma_{accel,y}^2$ y $\sigma_{accel,z}^2$. Para la simulación realiza el acelerómetro que se ha utilizado corresponde al acelerómetro Analog Devices ADXL325.

5.1.2 Giróscopo

Un giróscopo es un sensor mecánico basado en el Principio de Coriolis, que mide la velocidad angular de la aeronave con respecto a un sistema de referencia inercial. Es un aparato en el cual una masa que gira velozmente alrededor de su eje de simetría permite mantener de forma constante su orientación respecto a un sistema de ejes de referencia. El rápido movimiento giratorio del rotor de los giróscopos se puede obtener

por vacío o por un sistema eléctrico. El modelo matemático del giróscopo es el siguiente:

$$y_{gyro,x} = p + \eta_{gyro,x} \quad (5.2a)$$

$$y_{gyro,y} = q + \eta_{gyro,y} \quad (5.2b)$$

$$y_{gyro,z} = r + \eta_{gyro,z} \quad (5.2c)$$

donde el ruido presente en los sensores es modelado mediante los parámetros $\eta_{gyro,x}$, $\eta_{gyro,y}$ y $\eta_{gyro,z}$, que corresponden a ruido gaussiano con covarianzas $\sigma_{gyro,x}^2$, $\sigma_{gyro,y}^2$ y $\sigma_{gyro,z}^2$. Para la simulación realiza el acelerómetro que se ha utilizado corresponde al acelerómetro Analog Devices ADXRS450.

5.1.3 Altímetro barométrico

Un altímetro barométrico es un sensor que se encarga de obtener la altitud de vuelo a partir de la medida de presión atmosférica. Un sensor de presión absoluta se encarga de medir la presión de la atmósfera, utilizando para ello el Modelo de Atmósfera ISA Internacional. El modelo matemático empleado es el siguiente:

$$P_{barométrica} [Pa] = P_0 \left[\frac{T_0}{T_0 + L_0 h_{ASL}} \right]^{\frac{gM}{RL_0}} \quad (5.3a)$$

$$y_{abs\ pres} = (P_0 - P_{barométrica}) + \beta_{abs\ pres} + \eta_{abs\ pres} \quad (5.3b)$$

donde el ruido presente en los sensores es modelado mediante el parámetro $\eta_{abs\ pres}$, que corresponde a ruido gaussiano con covarianza $\sigma_{abs\ pres}^2$. El término $\beta_{abs\ pres}$ corresponde a un offset de temperatura.

La expresión (5.4) permite calcular el valor de altura en metros sobre un punto de referencia a partir de la salida del sensor de presión absoluta:

$$h_{barométrica} [m] = 0.3048 \cdot \frac{(1 - (P_{barométrica}/P_0)^{0.19026}) \cdot 288.15}{0.00198122} \quad (5.4)$$

5.1.4 Anemómetro

El anemómetro es un dispositivo que se encarga de medir la velocidad del aire a partir de un tubo de pitot con un sensor de presión diferencial. Este sensor de presión diferencial mide la diferencia entre la presión absoluta y presión estática y mediante señales eléctricas es capaz de obtener el valor de la presión diferencia. El modelo matemático empleado es:

$$y_{diff\ pres} = \frac{\rho V_a^2}{2} + \beta_{diff\ pres} + \eta_{diff\ pres} \quad (5.5)$$

donde el ruido presente en los sensores es modelado mediante el parámetro $\eta_{diff\ pres}$, que corresponde a ruido gaussiano con covarianza $\sigma_{diff\ pres}^2$. El término $\beta_{diff\ pres}$ corresponde a un offset de temperatura. A partir de la salida del sensor de presión diferencial se calcula la velocidad del aire. Para la simulación se ha utilizado al anemómetro Freescale Semiconductor MPXV5004G.

5.1.5 Magnetómetro

Un magnetómetro mide la intensidad y dirección del campo magnético en el sistema de referencia ejes cuerpo. Si se conoce el campo magnético de la tierra en ejes fijos, se puede obtener la actitud comparándola con la medida del instrumento. El modelo matemático empleado es el siguiente:

$$\phi_{mag} = \phi + \beta_{mag} + \eta_{mag} \quad (5.6a)$$

$$\theta_{mag} = \theta + \beta_{mag} + \eta_{mag} \quad (5.6b)$$

$$\psi_{mag} = \psi + \beta_{mag} + \eta_{mag} \quad (5.6c)$$

donde el ruido presente en los sensores es modelado mediante el parámetro η_{mag} , que corresponde a ruido gaussiano con covarianza σ_{mag}^2 . El término β_{mag} corresponde a un error de bias. Para la simulación se ha utilizado el Magnetómetro Honeywell HMR3300.

5.2 Sistema GPS

El Sistema de posicionamiento global (GPS) es un sistema de navegación por satélite que proporciona información acerca de la posición de un determinado cuerpo en la superficie terrestre. Está formado por una constelación de 24 satélites que orbitan continuamente alrededor de la Tierra a una altitud de 20180 km. Al medir los tiempos de vuelo de las señales desde un mínimo de cuatro satélites a un receptor en la superficie de la Tierra o cerca de ella, se puede determinar la ubicación del receptor en tres dimensiones.

La precisión de una medición de posición GPS se ve afectada por la precisión de las mediciones de pseudodistancia del satélite y por la geometría de los satélites a partir de las cuales se toman las mediciones de pseudodistancia. La precisión de pseudodistancia se ve afectada por errores en el tiempo de medición de vuelo para cada satélite. Dado que las señales de radio electromagnéticas de los satélites viajan a la velocidad de la luz, pequeños errores de tiempo pueden causar errores de posicionamiento significativos. Por ejemplo, un error de tiempo de solo 10 ns puede resultar en un error de posicionamiento de unos 3 m.

Para los propósitos de la simulación también hay que tener en cuenta la dinámica del error cometido por el Sistema GPS. Los errores de norte, este y altitud se componen de un sesgo que cambia lentamente junto con ruido aleatorio. Para modelar el comportamiento transitorio del error, seguimos el enfoque de [37] y modelamos el error como un proceso de Gauss-Markov. Los procesos de Gauss-Markov son modelados por

$$v[n+1] = e^{-k_{GPS}TS}v[n] + \eta_{GPS}[n] \quad (5.7)$$

donde $v[n]$ es el error que se está modelando, $\eta_{GPS}[n]$ es ruido blanco Gaussiano de media cero, $1/k_{GPS}$ es la constante de tiempo del proceso y TS es el tiempo de muestreo.

Gauss-Markov error model parameters

Direction	Nominal 1- σ error (m)		Model Parameters		
	Bias	Random	Std. Dev. η_{GPS} (m)	$1/k_{GPS}$ (s)	T_s (s)
North	4.7	0.4	0.21	1100	1.0
East	4.7	0.4	0.21	1100	1.0
Altitude	9.2	0.7	0.40	1100	1.0

Figura 5.1 Parámetros del modelo de error de Gauss-Markov..

Un modelo para mediciones GPS que es adecuado para propósitos de simulación es dado por

$$y_{GPS,n}[n] = p_n[n] + v_n[n] \quad (5.8a)$$

$$y_{GPS,e}[n] = p_e[n] + v_e[n] \quad (5.8b)$$

$$y_{GPS,h}[n] = -p_d[n] + v_h[n] \quad (5.8c)$$

donde p_n, p_e es la posición real del UAV en la Tierra y h la altitud sobre el nivel del mar, y n es el índice de muestra.

Mediante el efecto Doppler presente en las señales de satélite GPS, la velocidad del receptor se puede calcular a precisiones con desviaciones estándar en el rango de 0.01 a 0.05 mx/s. Hay que tener en cuenta que

la incertidumbre en la medición del rumbo varía con la inversa de la velocidad de avance: para velocidades altas, el error es pequeño y para velocidades bajas el error es grande.

Podemos modelar la velocidad del terreno y las medidas del recorrido disponible desde GPS como

$$y_{GPS,V_g} = \sqrt{(V_a \cos(\psi) + w_n)^2 + (V_a \sin(\psi) + w_e)^2} + \eta_V \quad (5.9a)$$

$$y_{GPS,\chi} = \text{atan2}(V_a \sin(\psi) + w_e, V_a \cos(\psi) + w_n) + \eta_\chi \quad (5.9b)$$

donde w_n y w_e es la velocidad del viento en ejes NED y los parámetros η_V y η_χ son ruido gaussiano con varianzas $\sigma_{V_g}^2$ y σ_χ^2 .

5.3 Filtro de Kalman Extendido

El sistema que hemos modelado corresponde a un sistema no lineal continuo en el tiempo representado por ecuaciones diferenciales ordinarias, mientras que los sensores que hemos desarrollado son sistemas que toman medidas en intervalos de tiempo T_s , es decir, se trata de un sistema discreto. Por tanto, el observador que vamos a implementar estará definido en tiempo discreto.

Un sistema no lineal puede escribirse de la forma

$$\dot{x} = f(x, u, t) + \zeta$$

$$y[n] = h(x[n], u[n]) + \eta[n]$$

donde x representa el estado del sistema, u la entrada del sistema y ζ, η perturbaciones modeladas como ruido blanco.

En este apartado explicaremos cómo sabiendo el estado del sistema dado por las ecuaciones y los modelos anteriores, conocidas las entradas del sistema y las medidas (sensores) cada cierto instante de tiempo, podemos obtener una estimación del sistema, denotado por \hat{x} . Para ello hacemos uso del Filtro de Kalman: conocida $u(t)$ y una estimación inicial \hat{x}_0 , intentar reconstruir las variables de estado conociendo el modelo matemático y las medidas proveniente de los sensores cada cierto intervalo de tiempo. Al tratarse de un sistema no lineal es necesario hacer uso del Filtro de Kalman Extendido (EKF), que permite hacer estimaciones de sistemas discretos no lineales.

Supongamos que las ecuaciones de transición de estado y de medición para un sistema no lineal de tiempo discreto tienen términos de proceso y ruido de medición no aditivos con media cero y matrices de covarianza Q y R , respectivamente:

$$x[k+1] = f(x[k], w[k], u_s[k])$$

$$y[k] = h(x[k], v[k], u_m[k])$$

$$w[k] \sim (0, Q[k])$$

$$v[k] \sim (0, R[k])$$

El algoritmo que se encarga de implementar el EKF es el siguiente:

1. Inicializar el objeto de filtro con los valores iniciales del estado, $x[0]$, y la matriz de covarianza del error de estimación del estado, P .

$$\hat{x}[0|-1] = E(x[0])$$

$$P[0|-1] = E((x[0] - \hat{x}[0|-1])(x[0] - \hat{x}[0|-1])^T)$$

donde \hat{x} es el estado estimado y $\hat{x}[k_a|k_b]$ denota el estado estimado en el paso de tiempo k_a usando medidas en instantes $0, 1, \dots, k_b$. Así, $\hat{x}[0|-1]$ es la mejor estimación del valor del estado antes de realizar cualquier medición. Este valor es necesario proporcionarlo al comenzar el algoritmo.

2. Para pasos de tiempo $k = 0, 1, 2, 3, \dots$ realizar los siguientes pasos:

a) Calcular el jacobiano de la función de medición y actualice la covarianza del error de estimación de estado y estado utilizando los datos medidos, $y[k]$.

$$C[k] = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}[k|k-1]}$$

$$S[k] = \left. \frac{\partial h}{\partial v} \right|_{\hat{x}[k|k-1]}$$

Calcular estas matrices jacobianas numéricamente a menos que se especifique el jacobiano analítico.

$$K[k] = P[k|k-1]C[k]^T (C[k]P[k|k-1]C[k]^T + S[k]R[k]S[k]^T)^{-1}$$

$$\hat{x}[k|k] = \hat{x}[k|k-1] + K[k] (y[k] - h(\hat{x}[k|k-1], 0, u_m[k]))$$

$$\hat{P}[k|k] = P[k|k-1] - K[k]C[k]P[k|k-1]$$

donde K es la ganancia de Kalman.

b) Calcular el jacobiano de la función de transición de estado y la predicción de la covarianza del error de estimación de estado y el estado en el siguiente paso de tiempo.

$$A[k] = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}[k|k]}$$

$$G[k] = \left. \frac{\partial f}{\partial w} \right|_{\hat{x}[k|k]}$$

Calcular estas matrices jacobianas numéricamente a menos que especifique el jacobiano analítico.

$$P[k+1|k] = A[k]P[k|k]A[k]^T + G[k]Q[k]G[k]^T$$

$$\hat{x}[k+1|k] = f(\hat{x}[k|k], 0, u_s[k])$$

Los pasos del algoritmo descritos anteriormente suponen que tiene términos de ruido no aditivos en las funciones de transición de estado y medición. Si hay términos de ruido aditivo en las funciones, los cambios en el algoritmo son:

- Si el ruido del proceso w es aditivo, es decir, la ecuación de transición de estado tiene la forma $x[k] = f(x[k-1], u_s[k-1] + w[k-1])$, entonces la matriz jacobiana $G[k]$ es una matriz identidad.
- Si el ruido de medición v es aditivo, es decir, la ecuación de medición tiene la forma $y[k] = h(x[k], u_m[k] + v[k])$, entonces la matriz jacobiana $S[k]$ es una matriz identidad.

5.3.1 Filtro de Kalman Extendido para Posición

Aplicamos el EKF desarrollado en este apartado para poder realizar una estimación sobre la posición del UAV a partir de las medidas proporcionadas por el Sistema GPS. Las ecuaciones que permiten conocer la derivada del vector posición en ejes NED viene dado por (2.1).

Expresando estas ecuaciones en la forma necesaria para aplicar el EKF:

$$p_n[k+1] = p_n[k] + \left[u[k]c_{\theta[k]}c_{\psi[k]} + v[k](s_{\phi[k]}s_{\theta[k]}c_{\psi[k]} - c_{\phi[k]}s_{\psi[k]}) + w[k](c_{\phi[k]}s_{\theta[k]}c_{\psi[k]} + s_{\phi[k]}s_{\psi[k]}) \right] dt \quad (5.12a)$$

$$p_e[k+1] = p_e[k] + \left[u[k]c_{\theta[k]}s_{\psi[k]} + v[k](s_{\phi[k]}s_{\theta[k]}s_{\psi[k]} + c_{\phi[k]}c_{\psi[k]}) + w[k](c_{\phi[k]}s_{\theta[k]}s_{\psi[k]} - s_{\phi[k]}c_{\psi[k]}) \right] dt \quad (5.12b)$$

$$p_d[k+1] = p_d[k] + \left[-u[k]s_{\theta[k]} + v[k](s_{\phi[k]}c_{\theta[k]}) + w[k](c_{\phi[k]}c_{\theta[k]}) \right] dt \quad (5.12c)$$

donde se ha empleado la notación $c_\alpha \equiv \cos(\alpha)$, $s_\alpha \equiv \sin(\alpha)$. El vector de estados corresponde a $x = (p_n, p_e, p_d)^T$ y las entradas del sistema $u = (u, v, w, \phi, \theta, \psi)^T$. Las medidas $y[k]$ del sistema GPS vienen dadas por la expresión (5.8).

5.3.2 Filtro de Kalman Extendido para Actitud

Aplicamos el EKF desarrollado en este apartado para poder realizar una estimación sobre la actitud del UAV a partir de las medidas proporcionadas por el magnetómetro. Las ecuaciones que permiten conocer la derivada del vector posición en ejes NED viene dado por (2.2).

Expresando estas ecuaciones en la forma necesaria para aplicar el EKF:

$$\dot{\phi}[k+1] = \dot{\phi}[k] + [p[k] + q[k] \sin(\phi[k]) \tan(\theta[k]) + r \cos(\phi[k]) \tan(\theta[k])] dt \quad (5.13a)$$

$$\dot{\theta}[k+1] = \dot{\theta}[k] + [q[k] \cos(\phi[k]) - r[k] \sin(\phi[k])] dt \quad (5.13b)$$

$$\dot{\psi}[k+1] = \dot{\psi}[k] + [q[k] \sin(\phi[k]) \sec(\theta[k]) + r[k] \cos(\phi[k]) \sec(\theta[k])] dt \quad (5.13c)$$

donde el vector de estados corresponde a $x = (\phi, \theta, \psi)^T$ y las entradas del sistema $u = (p, q, r)^T$. Las medidas $y[k]$ del magnetómetro vienen dadas por la expresión (5.6).

6 Implementación en Matlab-Simulink

En este capítulo se van a realizar las simulaciones correspondientes al UAV modelado en los capítulos anteriores mediante la herramienta *Matlab-Simulink*. Antes de ir a la presentación de resultados se va a hacer una breve referencia sobre los distintos bloques empleados en Simulink y cómo han sido configurados.

6.1 Bloque de Cinemática y Dinámica

Este bloque se encarga del cálculo de la cinemática y dinámica del UAV a partir de las ecuaciones planteadas en el Capítulo 2.1.

6.1.1 Cinemática

Para el cálculo de la cinemática del UAV se ha hecho uso de dos bloques: uno que se encarga de hallar las componentes de la velocidad lineal y otro de la velocidad angular.

El bloque de cinemática lineal cuenta con 6 variables de entrada: $u, v, w, \phi, \theta, \psi$ y 3 variables de salida: $\dot{p}_n, \dot{p}_e, \dot{p}_d$. Para hallar la posición de la aeronave se hace pasar la salida del bloque por un integrador para así obtener la posición del UAV en cada instante de simulación.

El bloque de cinemática angular cuenta con 5 entradas: p, q, r, ϕ, θ y 3 variables de salida: $\dot{\phi}, \dot{\theta}, \dot{\psi}$. Haciendo pasar dicha salida por un integrador se obtiene la actitud del UAV.

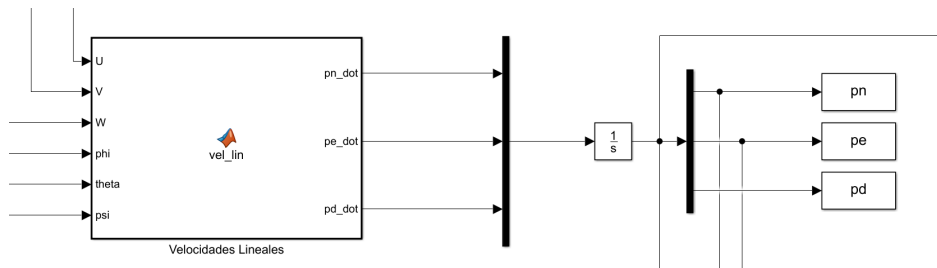


Figura 6.1 Bloque Cinemática lineal en Simulink..

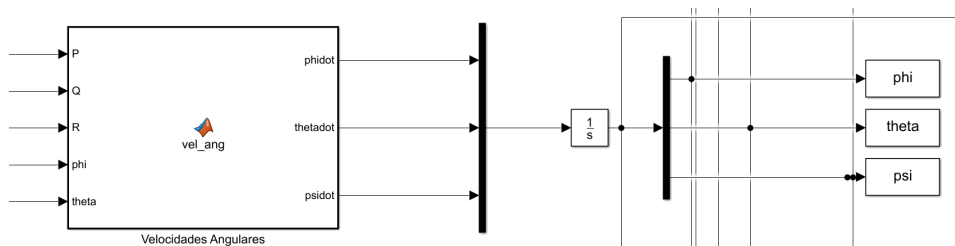


Figura 6.2 Bloque Cinemática angular en Simulink..

6.1.2 Dinámica

Para el cálculo de la cinemática del UAV se ha hecho uso de dos bloques: uno que se encarga de hallar las componentes de la aceleración lineal y otro de la aceleración angular.

El bloque de aceleración lineal cuenta con 9 entradas: $u, v, w, p, q, r, f_x, f_y, f_z$ y 3 variables de salida: $\dot{u}, \dot{v}, \dot{w}$. Haciendo pasar dicha salida por un integrador se obtiene la velocidad del UAV.

El bloque de aceleración angular cuenta con 6 entradas: p, q, r, l, m, n y 3 variables de salida: $\dot{u}, \dot{v}, \dot{w}$. Haciendo pasar dicha salida por un integrador se obtiene la velocidad angular del UAV.

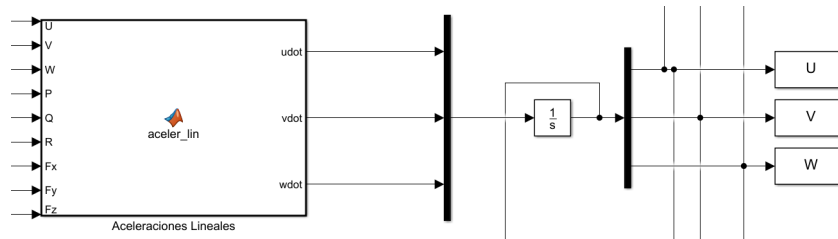


Figura 6.3 Bloque Dinámica lineal en Simulink..

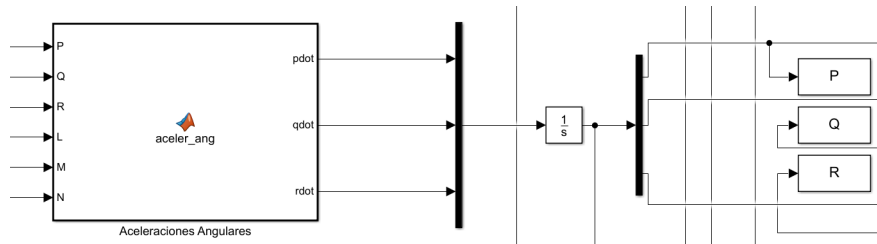


Figura 6.4 Bloque Dinámica angular en Simulink..

6.2 Bloque de Fuerzas y Momentos

En este apartado se muestran los bloques empleados para el cálculo de las fuerzas y momentos que actúan sobre el UAV. También se hará un breve comentario sobre la obtención de los coeficientes aerodinámicos y modelos gravitatorio y de viento.

6.2.1 Modelo gravitatorio

Para el modelo de gravedad se ha hecho uso del bloque WGS84 de Simulink. Se encarga de obtener el valor de la gravedad a partir de la posición del UAV en coordenadas geodésicas. Por lo tanto, es necesario pasar de coordenadas NED a geodésicas antes de entrar al bloque.

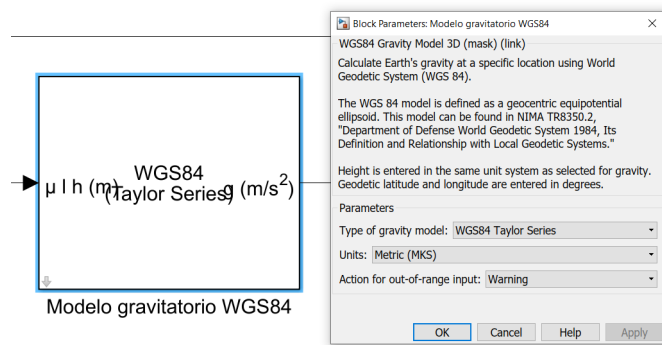


Figura 6.5 Bloque WGS84 de Simulink..

6.2.2 Modelo aerodinámico

Para el cálculo de los coeficientes aerodinámicos del UAV se hace uso de un bloque con 9 entradas: $u, v, w, p, q, r, \delta_a, \delta_e, viento(u_v, v_v, w_v, p_v, q_v, r_v)$. Los coeficientes de salida son 6: $C_L, C_D, C_Y, C_l, C_m, C_n$.

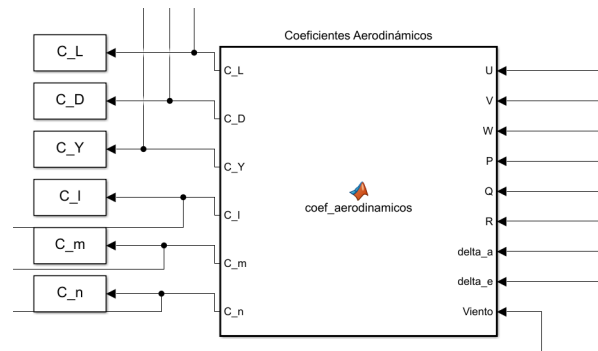


Figura 6.6 Bloque Coeficientes Aerodinámicos en Simulink..

6.2.3 Viento

Este bloque se encarga de modelar el viento presente mediante el modelo de viento continuo de Dryden. El sistema tiene 7 entradas: $h, u, v, w, \phi, \theta, \psi$ y como salidas las componentes del viento: $u_v, v_v, w_v, p_v, q_v, r_v$ y la velocidad relativa del UAV respecto a la masa de aire circundante: u_a, v_a, w_a .

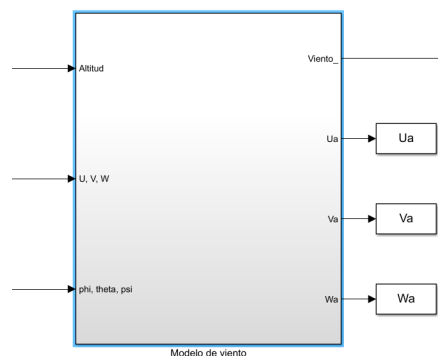


Figura 6.7 Bloque de Modelado del Viento en Simulink..

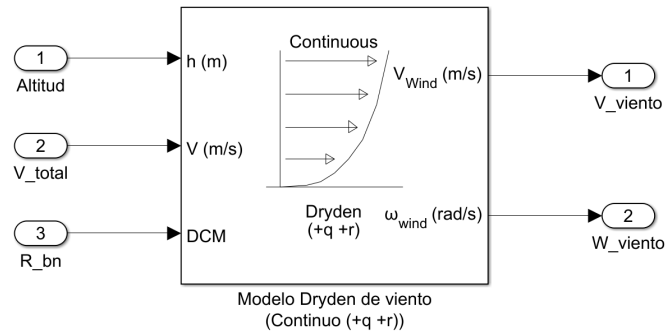


Figura 6.8 Bloque de Dryden de Simulink..

Para la configuración del modelo de Dryden se ha hecho uso de la rosa de los vientos correspondiente a Sevilla, la cual podemos ver en las imágenes (6.9) y (6.10). Se ha decidido utilizar para la simulación un viento de 10 m/s y componente WSW (247.5°).

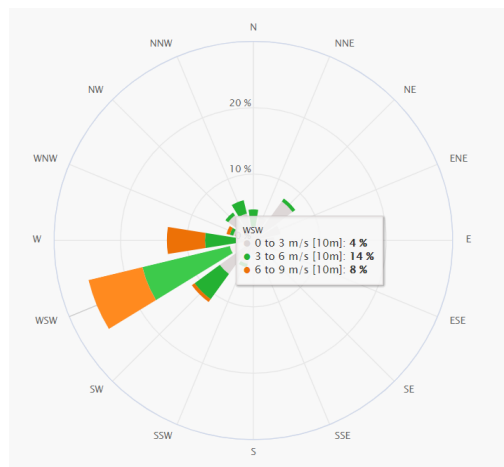


Figura 6.9 Viento predominante a 10 m de altitud..

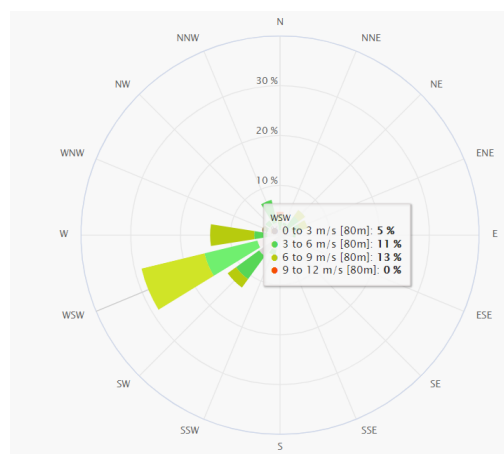


Figura 6.10 Viento predominante a 80 m de altitud..

6.2.4 Fuerzas y Momentos

El bloque encargado de calcular las fuerzas que experimenta el UAV durante la simulación se encuentra en la Fig. (6.11). A partir de las entradas que recibe calcula las fuerzas que experimenta en ejes cuerpo.

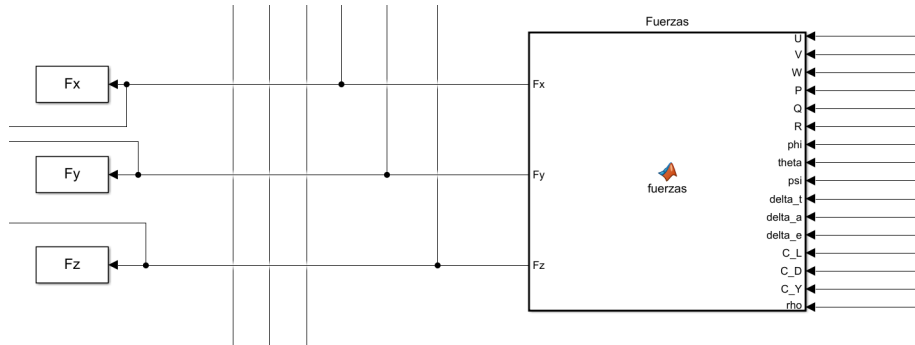


Figura 6.11 Bloque de Fuerzas en Simulink..

El bloque encargado de calcular los momentos que experimenta el UAV durante la simulación se encuentra en la Fig. (6.12).

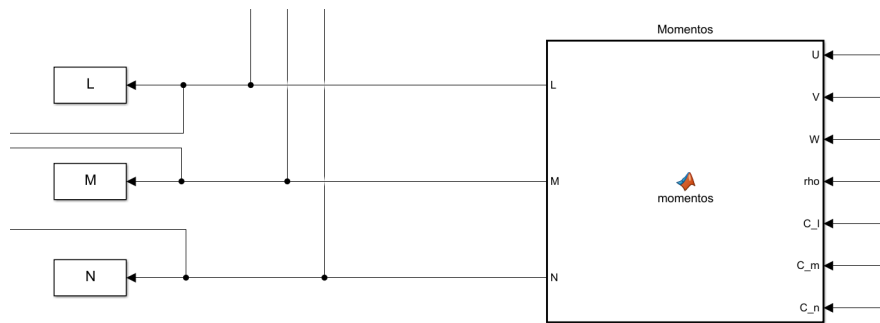


Figura 6.12 Bloque de Momentos en Simulink..

6.3 Controladores

Este bloque se encarga de implementar los controladores necesarios para el vuelo del UAV. Su objetivo es calcular el valor de las variables de control del sistema necesarias para el bloque de fuerzas y momentos, y así el UAV pueda llevar a cabo la misión objetivo.

6.3.1 Máquina de estados de Control de Altitud

Antes de mostrar el bloque correspondiente a cada controlador es de interés presentar el bloque que se encarga del modelado de la máquina de estados de control longitudinal. A partir de las entradas h , h_{hold} , $h_{takeoff}$ y h^c calcula en qué régimen de vuelo se encuentra ((3.6)) y por tanto el estado de cada controlador.

6.3.2 Controlador de Cabeceo

Implementa el controlador necesario para el cabeceo del UAV. A partir del valor de referencia se encarga de calcular la deflexión necesaria de los elevadores.

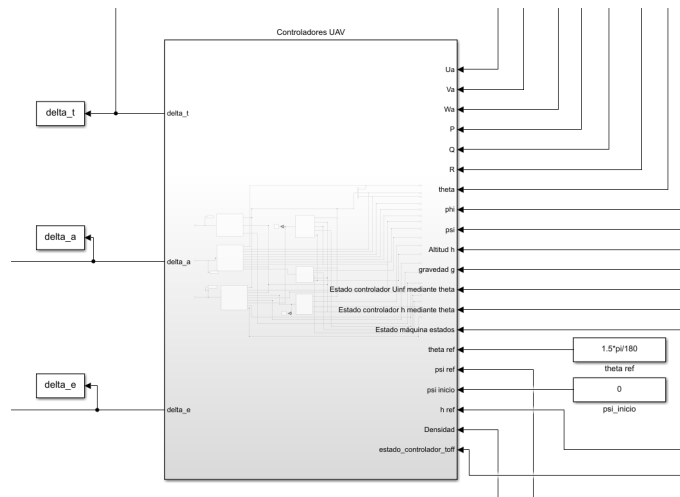


Figura 6.13 Bloque de controladores en Simulink..

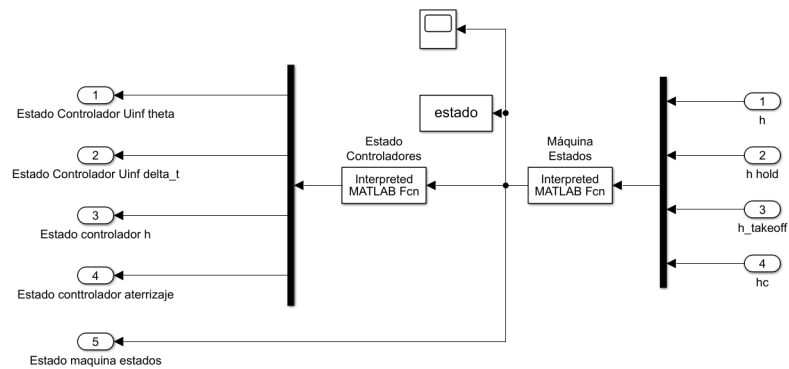


Figura 6.14 Máquina de estados de control longitudinal en Simulink..

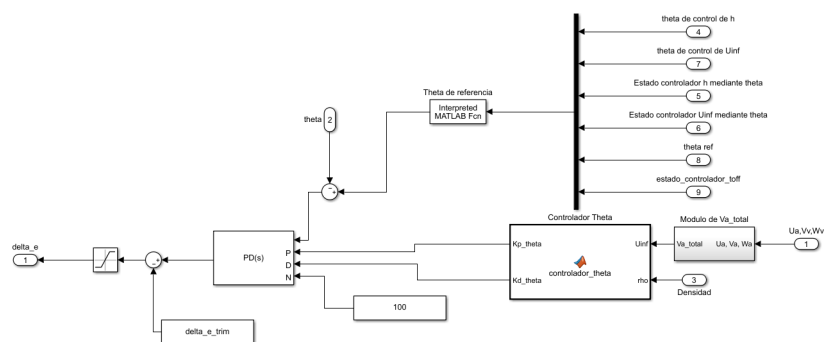


Figura 6.15 Controlador de Cabeceo en Simulink..

6.3.3 Controlador de Altitud

Implementa el controlador necesario para la altitud del UAV. A partir del valor de altitud de referencia se encarga de calcular el ángulo de cabeceo de referencia para el controlador de cabeceo.

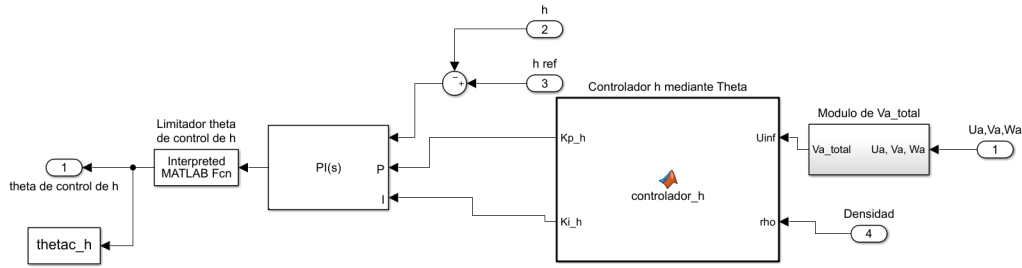


Figura 6.16 Controlador de Altitud en Simulink..

6.3.4 Controlador de Velocidad mediante Cabeceo

Implementa el controlador necesario para la velocidad del UAV mediante el ángulo de cabeceo. A partir del valor de velocidad de referencia se encarga de calcular el ángulo de cabeceo de referencia para el controlador de cabeceo.

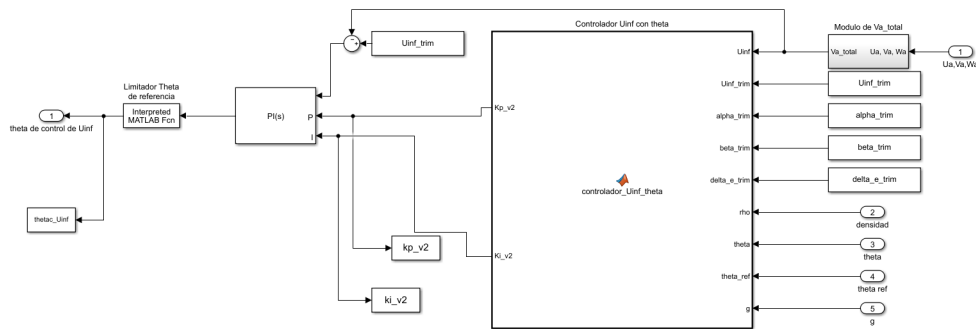


Figura 6.17 Controlador de Velocidad mediante Cabeceo en Simulink..

6.3.5 Controlador de Velocidad mediante Empuje

Implementa el controlador necesario para la velocidad del UAV mediante la señal PWM δ_t . A partir del valor de velocidad de referencia se encarga de calcular dicho valor de δ_t .

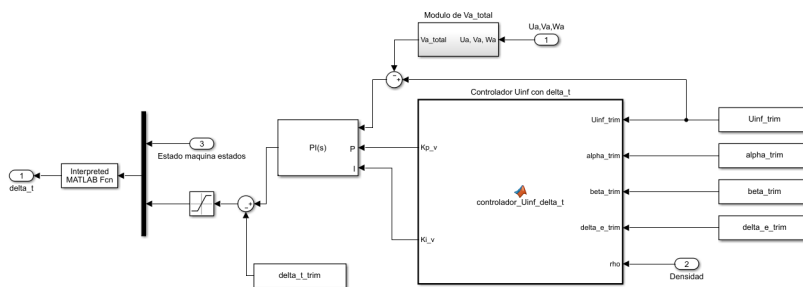


Figura 6.18 Controlador de Velocidad mediante Empuje en Simulink..

6.3.6 Controlador de Alabeo

Implementa el controlador necesario para el alabeo del UAV. A partir del valor de referencia se encarga de calcular la deflexión necesaria de los alerones.

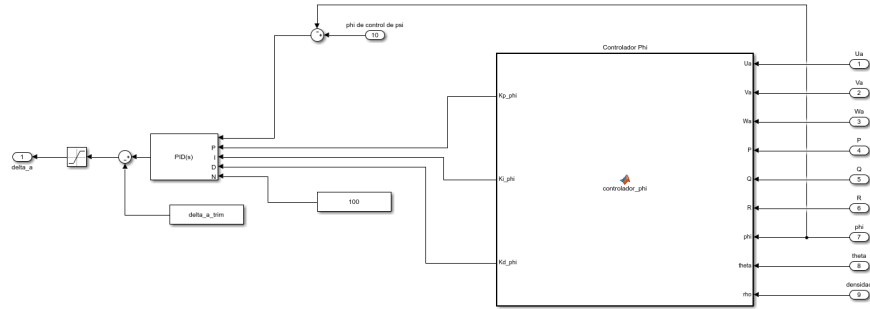


Figura 6.19 Controlador de Alabeo en Simulink..

6.3.7 Controlador de Rumbo

Implementa el controlador necesario para la guiñada del UAV. A partir del valor de referencia se encarga de calcular el ángulo de alabeo de referencia para el controlador de alabeo.

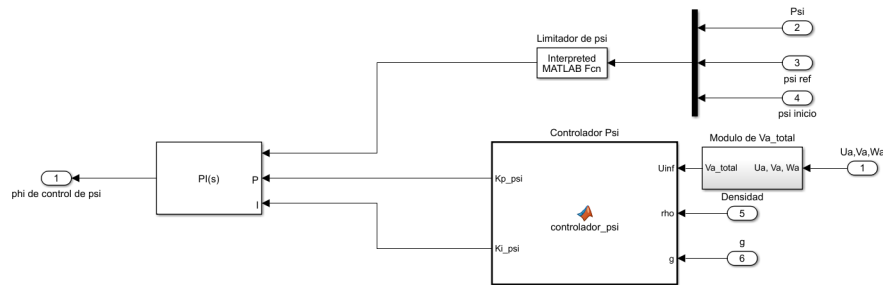


Figura 6.20 Controlador de Rumbo en Simulink..

6.4 Misión

Este bloque se encarga de calcular el valor de la altitud y rumbo de referencias para el guiado del UAV. Además también se define como variables de salida las coordenadas del waypoint de destino $(r_n, r_e, r_d)^T$ y un contador que se actualiza cada vez que el UAV pasa por dicho punto de destino.

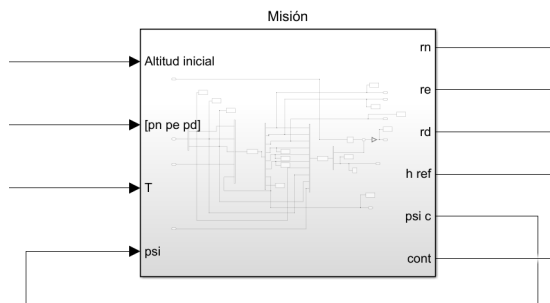


Figura 6.21 Sistema de misión en Simulink..

6.5 Sensores

6.5.1 Acelerómetro

Este bloque implementa el acelerómetro desarrollado en el Capítulo 5.1.1 y consta de 12 entradas: \dot{u} , \dot{v} , \dot{w} , u , v , w , p , q , r , ϕ , θ , g . Las salidas del bloque corresponden a las aceleraciones que experimenta el UAV: $y_{accel,x}$, $y_{accel,y}$, $y_{accel,z}$.

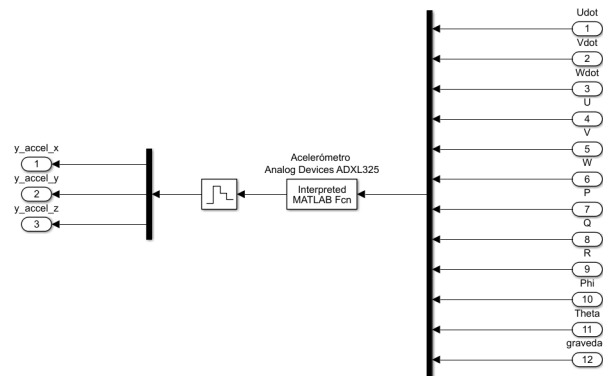


Figura 6.22 Acelerómetro en Simulink..

6.5.2 Giróscopo

Este bloque implementa el giróscopo desarrollado en el Capítulo 5.1.2 y consta de 3 entradas: p , q , r . Las salidas del bloque corresponde a la velocidad angular de el UAV: p_{gyro} , q_{gyro} , r_{gyro} .

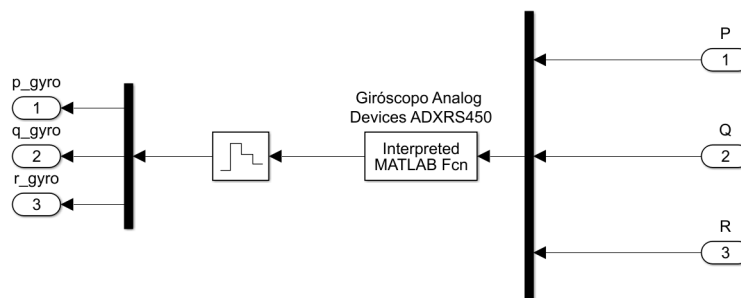


Figura 6.23 Giróscopo en Simulink..

6.5.3 Altimetro barométrico

Este bloque implementa el giróscopo desarrollado en el Capítulo 5.1.3 y consta de 3 entradas: p , q , r . Las salidas del bloque corresponde a la velocidad angular de el UAV: p_{gyro} , q_{gyro} , r_{gyro} .

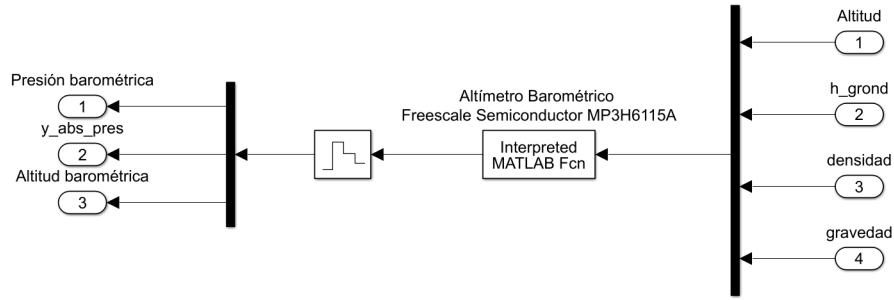


Figura 6.24 Altímetro barométrico en Simulink..

6.5.4 Anemómetro

Este bloque implementa el anemómetro desarrollado en el Capítulo 5.1.4 y consta de 2 entradas: ρ , V_a . La salida del bloque corresponde a la presión dinámica.

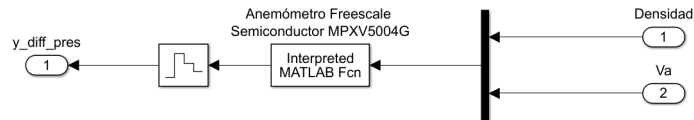


Figura 6.25 Anemómetro en Simulink..

6.5.5 Magnetómetro

Este bloque implementa el anemómetro desarrollado en el Capítulo 5.1.5 y consta de 3 entradas: ϕ , θ , ψ . La salida del bloque corresponde a la actitud del UAV: ϕ_{mag} , θ_{mag} , ψ_{mag} .

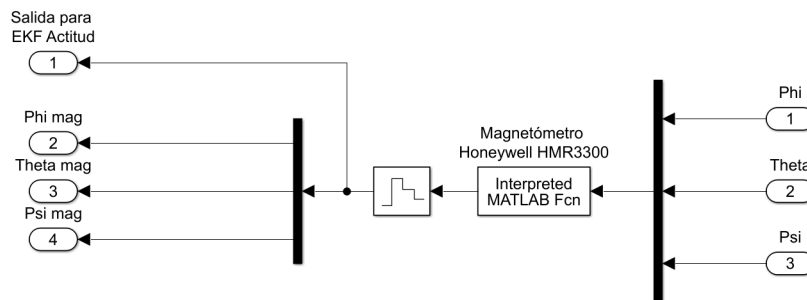


Figura 6.26 Magnetómetro en Simulink..

6.6 Sistema GPS

Bloque que implementa el Sistema GPS desarrollado en el Capítulo 5.2. A partir de las entradas al bloque se encarga de calcular la posición, velocidad y rumbo del UAV. Para ello se ha introducido una función que se encarga de calcular las componentes del viento en ejes NED: $(w_n, w_e, w_d)^T$.

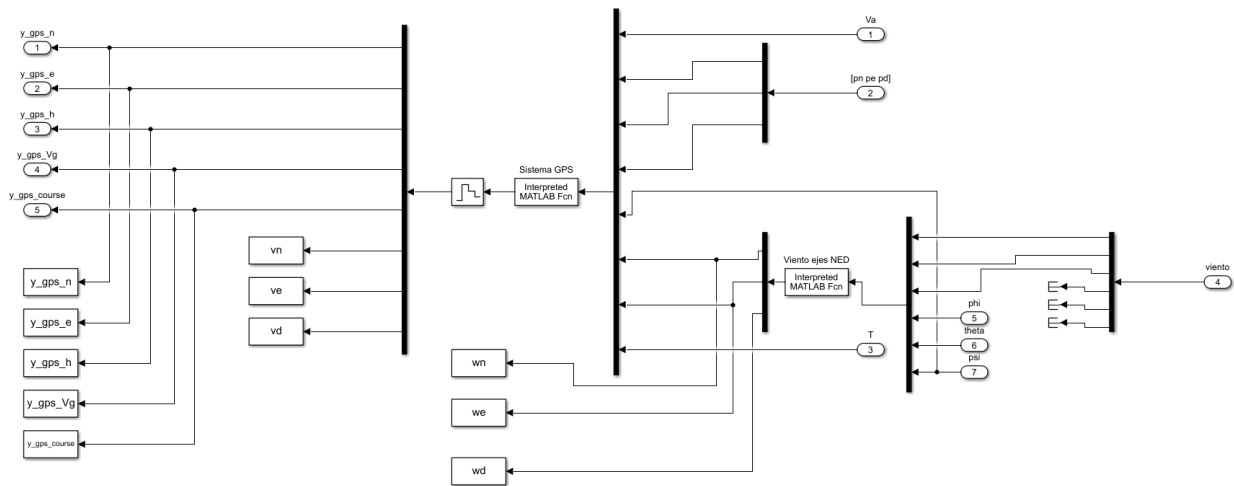


Figura 6.27 Sistema GPS en Simulink..

6.7 Filtro de Kalman Extendido

Para la implementación del EKF se ha hecho uso del bloque “*Extended Kalman Filter*” de Simulink. Este bloque se encarga de aplicar el EKF a partir de una serie de parámetros de entrada:

- State Transition: función $f(x[k], w[k], u_s[k])$ que modela la dinámica del UAV. Hay que indicar si el ruido es aditivo o no aditivo y proporcionar la matriz de covarianza.
- Inizalitation: vector de estado inicial con su respectiva covarianza.
- Measurement: función $h(x[k], v[k], u_m[k])$ que modela las medidas tomadas por el sensor. Hay que indicar si el ruido es aditivo o no aditivo y proporcionar la matriz de covarianza.
- Sample Time: tiempo de muestreo.

A partir de estos parámetros el bloque calcula las variables estimadas. Este bloque calcula el jacobiano de manera numérica, aunque se le puede introducir la expresión analítica en el caso de que se disponga de ella, lo cual hace que los resultados proporcionados sean más fiables y con menor tiempo de computación.

6.7.1 EKF Posición

EL EKF diseñado para estimar la posición del UAV hace uso de las medidas tomadas por el Sistema GPS. Podemos ver el EKF implementado en la Fig. (6.28)

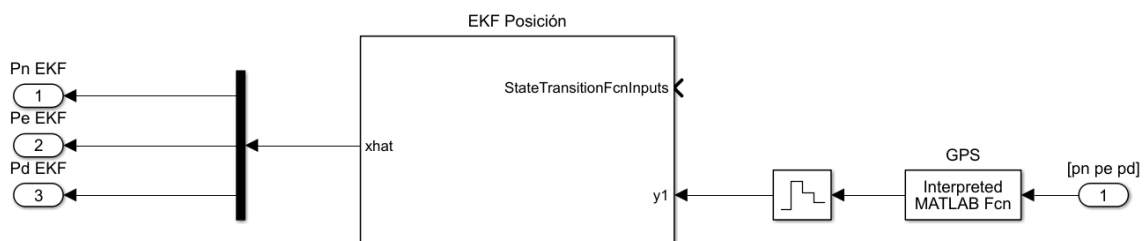


Figura 6.28 EKF de Posición en Simulink..

6.7.2 EKF Actitud

EL EKF diseñado para estimar la actitud del UAV hace uso de las medidas tomadas por el magnetómetro. Podemos ver el EKF implementado en la Fig. (6.29)

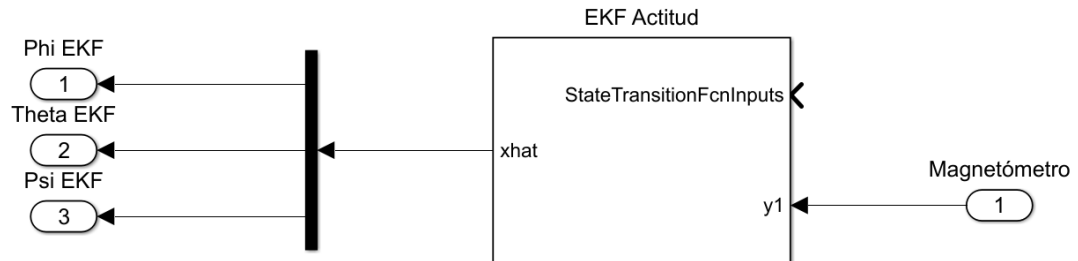


Figura 6.29 EKF de Actitud en Simulink..

7 Resultados

7.1 Misión realizada

En este Capítulo se van a mostrar los resultados obtenidos para la misión deseada. Ésta consiste en un vuelo que abarca una superficie de unos 200 hectáreas en las proximidades del Aeropuerto de Sevilla (LEZL (OACI), SVQ (IATA)), en el que el UAV recorre 6025m. Para simplificar la simulación se va considerar sólo la fase correspondiente a vuelo horizontal, obviando el despegue y aterrizaje. Se pretende que el UAV parta de un punto inicial y vuelva a dicho punto pasando por una serie de waypoints definidos. La simulación corresponde a un vuelo con velocidad de trimado del UAV de 15 m/s y viento de intensidad 10 m/s y componente WSW.

Además de la misión realizada por el UAV se van a mostrar otros resultados de interés, como la actitud de la aeronave durante toda la simulación, medidas tomadas por los sensores, la salida del EKF de posición y actitud, etc.

7.2 Trayectoria

En la Fig. (7.1) se muestra la trayectoria descrita por el UAV, obtenida mediante Matlab-Simulink. En ella se han representado los waypoints por los que se quiere que pase el UAV, el camino óptimo obtenido mediante el algoritmo TSP (color azul) y la trayectoria descrita por el UAV (color naranja). Hay que indicar que se ha decidido utilizar como origen de coordenadas el punto de partida de la aeronave $(0,0,100)^T$ por el mero hecho de facilitar el cálculo en Matlab. Posteriormente se calculará la posición geodésica real de los waypoints y puntos que conforman la trayectoria descrita por el UAV. Podemos comprobar cómo el UAV pasa por cada uno de los waypoints definidos y que la trayectoria que describe es prácticamente igual al camino óptimo. Las mayores diferencias las encontramos en aquellas maniobras que requieren virajes más bruscos, debido a que la aeronave está diseñada para realizar maniobras suaves. Aún así podemos afirmar que el modelo de guiado para el UAV cumple las expectativas con los que se comienza el proyecto.

Para dotar de mayor realismo a la simulación, en el Anexo A se muestra la misión llevada a cabo por el UAV en Google Earth. Se vuelve a representar los waypoints por los que se quiere que pase el UAV, el camino óptimo obtenido mediante el algoritmo TSP (color azul) y la trayectoria descrita por el UAV (color naranja).

También se van a representar los resultados de posición obtenidos mediante la integración de las ecuaciones diferenciales (ecs. 2.1) y las medidas tomadas por el Sistema GPS y la estimación del EKF de posición.

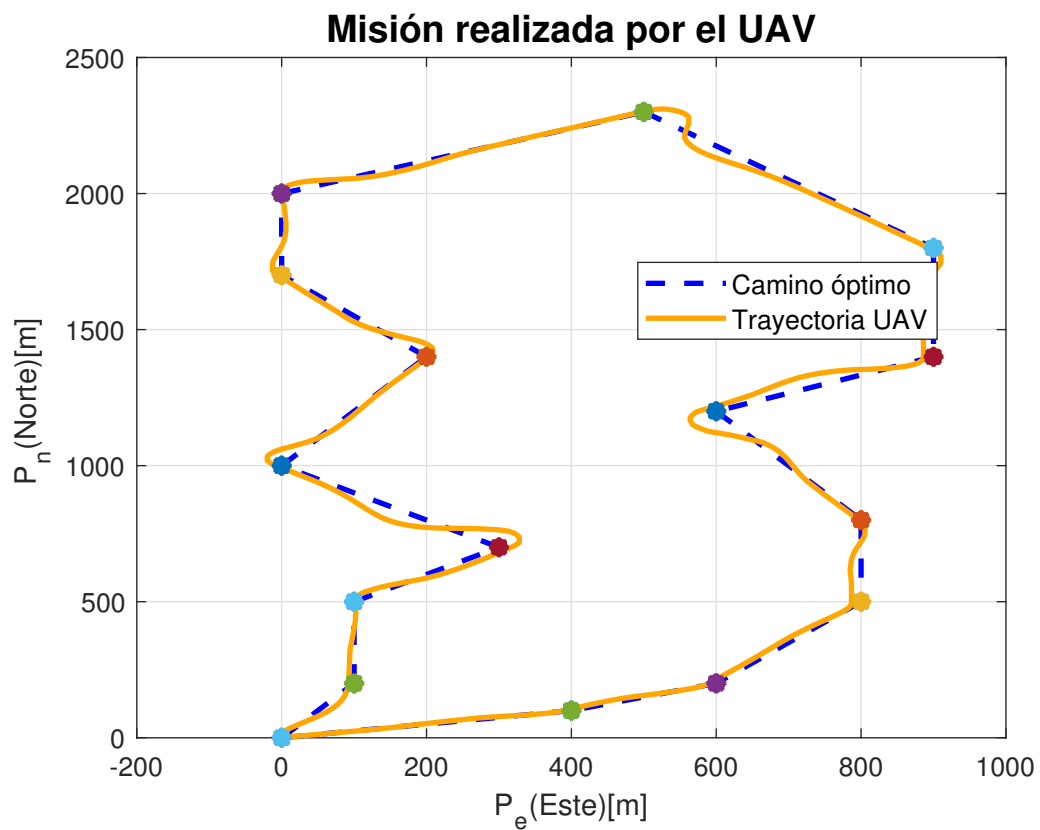


Figura 7.1 Misión realizada por el UAV en Matlab-Simulink..

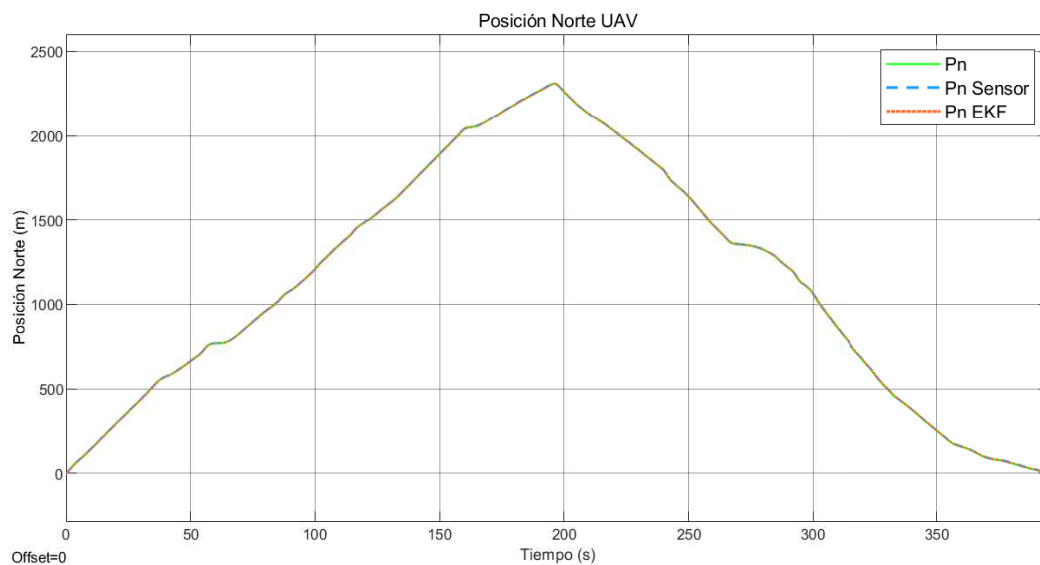


Figura 7.2 Posición Norte del UAV..

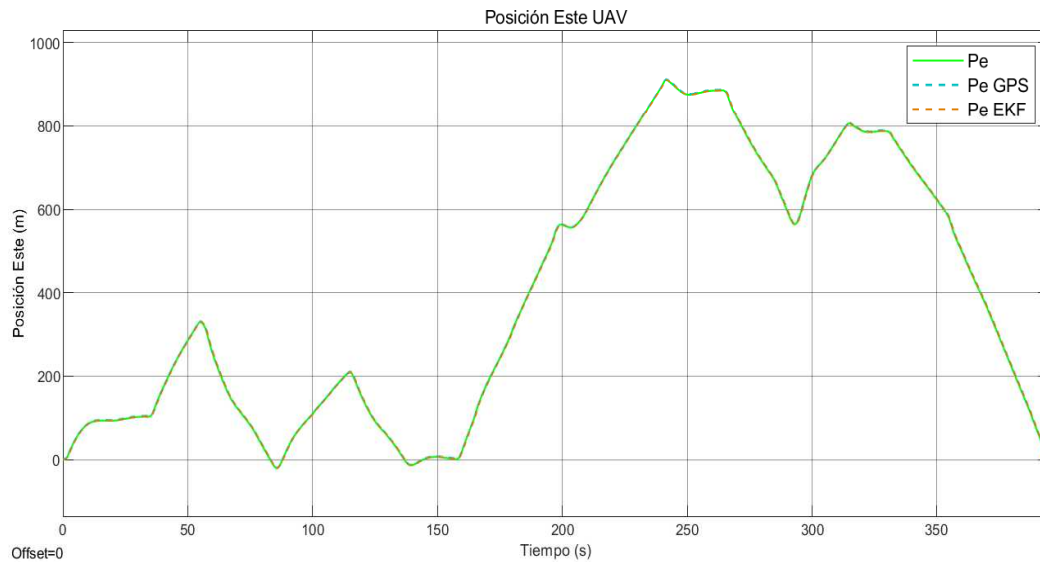


Figura 7.3 Posición Este del UAV..

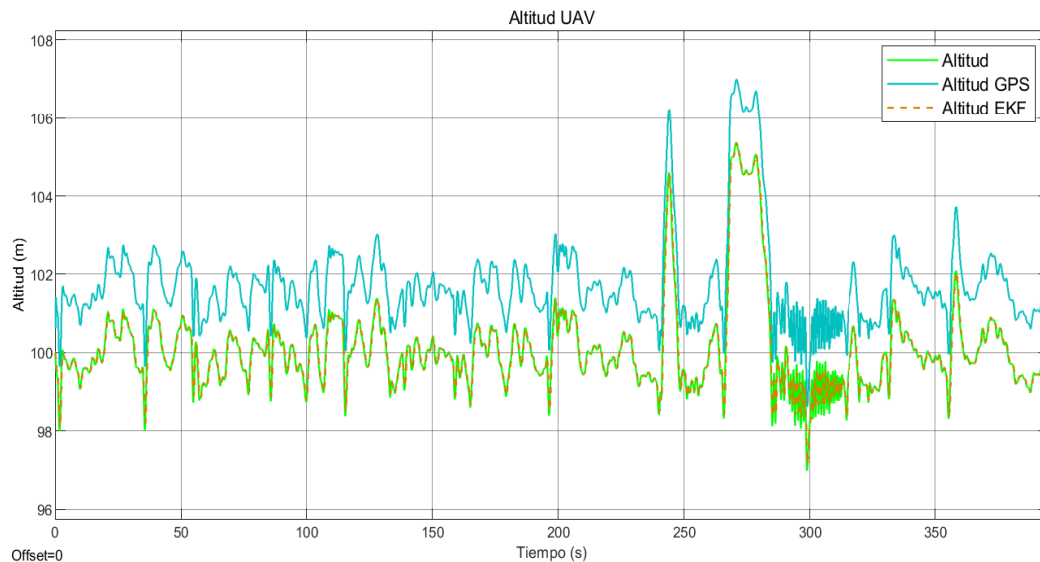


Figura 7.4 Altitud del UAV..

A la vista de los resultados, observamos como el EKF implementado para la estimación de la posición del UAV funciona correctamente. La mínima diferencia entre los valores reales y medidos/estimados se debe al error introducido por los sensores.

7.3 Actitud

En este apartado se van a mostrar los resultados correspondientes a la actitud del UAV. Para ello vamos a hacer uso del valor de ϕ , θ , ψ calculados mediante la integración de las ecuaciones diferenciales (ecs. 2.2), las medidas tomadas por el magnetómetro y la estimación del EKF de actitud.

Hay que indicar que los resultados obtenidos se esperan que tenga menor exactitud que los obtenidos anteriormente para posición, debido a que la aeronave se encuentra en todo momento cambiando su actitud para poder realizar la misión deseada.

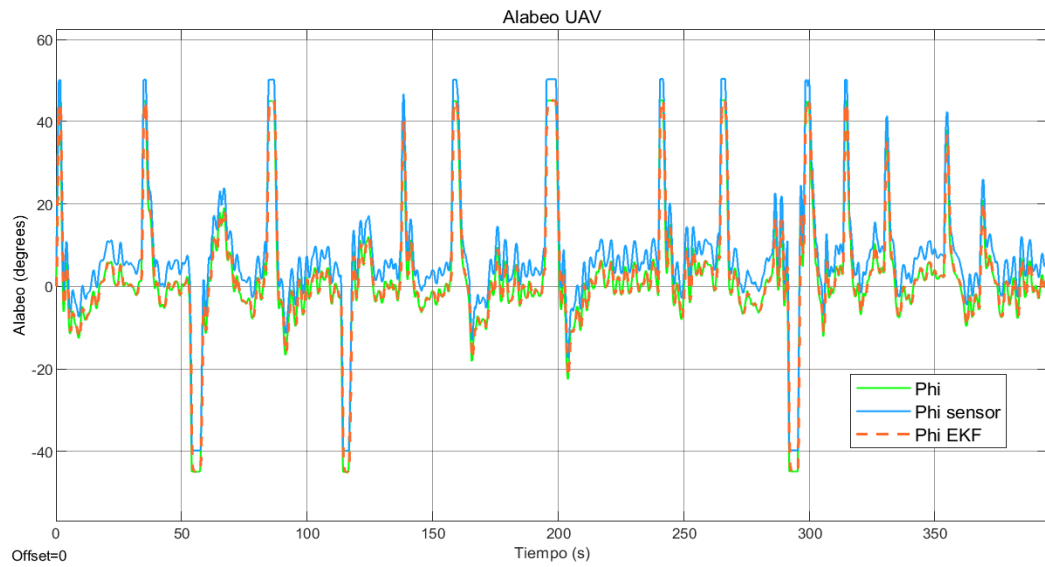


Figura 7.5 Alabeo del UAV..

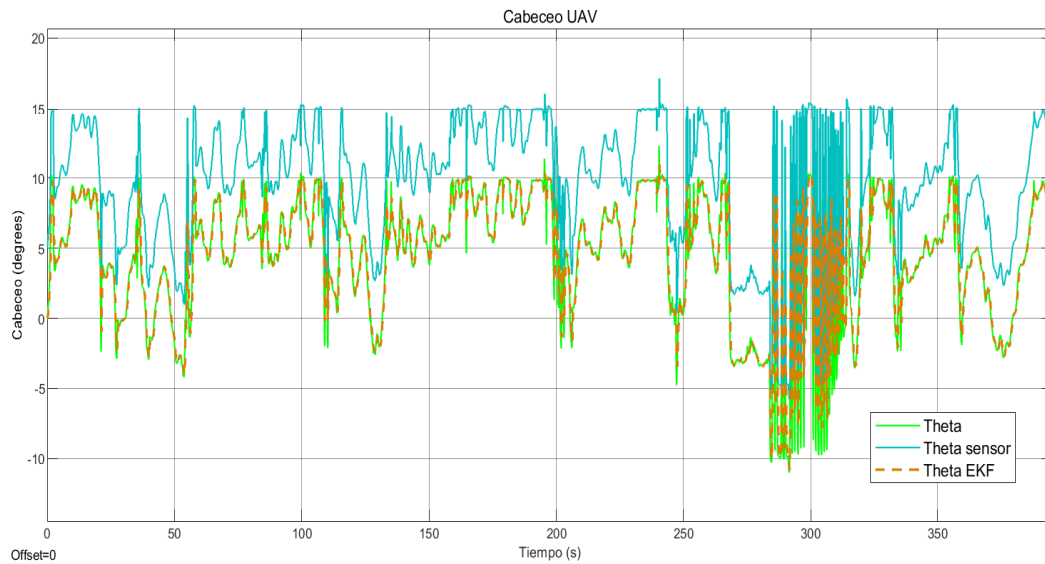


Figura 7.6 Cabeceo del UAV..

A la vista de los resultados, observamos como el EKF implementado para la estimación de la actitud del UAV funciona correctamente. La diferencia entre los valores reales y medidos/estimados se debe al error introducido por los sensores.

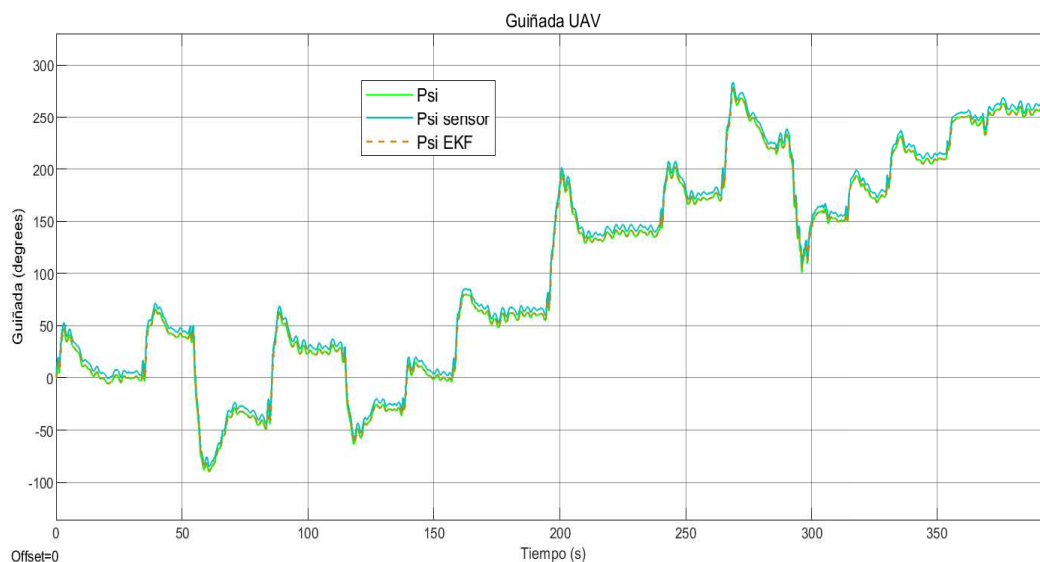


Figura 7.7 Guiñada del UAV..

7.4 Velocidad

En este apartado se muestran las 3 componentes de la velocidad del UAV durante la simulación. Comprobamos como la componente longitudinal es mayor que las demás componentes, algo totalmente lógico ya que corresponde a la dirección en la cual se produce la fuerza propulsiva. Ésta se encuentra en torno a 15 m/s , velocidad para la cual se había decidido trimar el UAV.

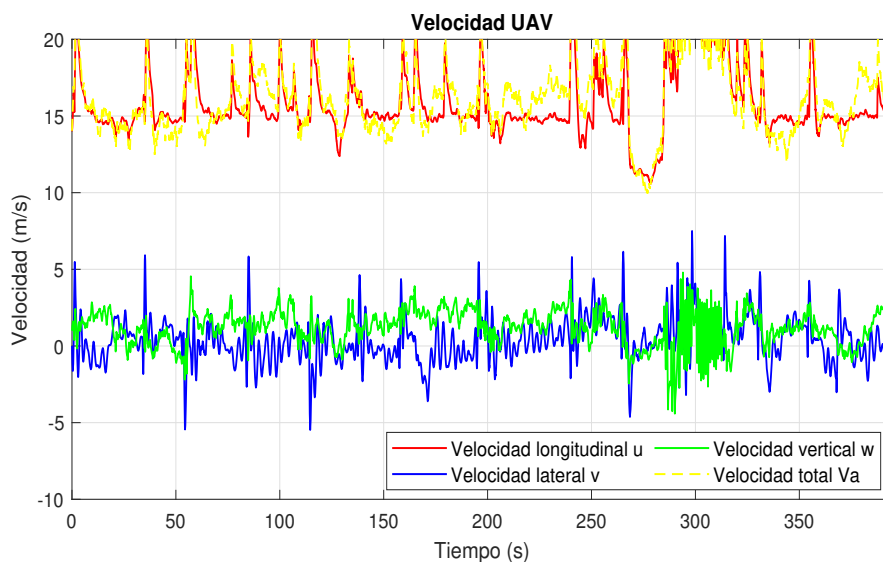


Figura 7.8 Velocidad del UAV..

7.5 Señal PWM de Empuje

En este apartado mostramos la evolución de la Señal PWM encargada de modelar el empuje del UAV. En ella podemos ver cómo durante la fase de retención de altitud, se ha limitado la señal al intervalo $[0.3, 0.4]$, valores dentro de los cuales se encuentra el valor de la Señal PWM para el trimado del UAV. También podemos

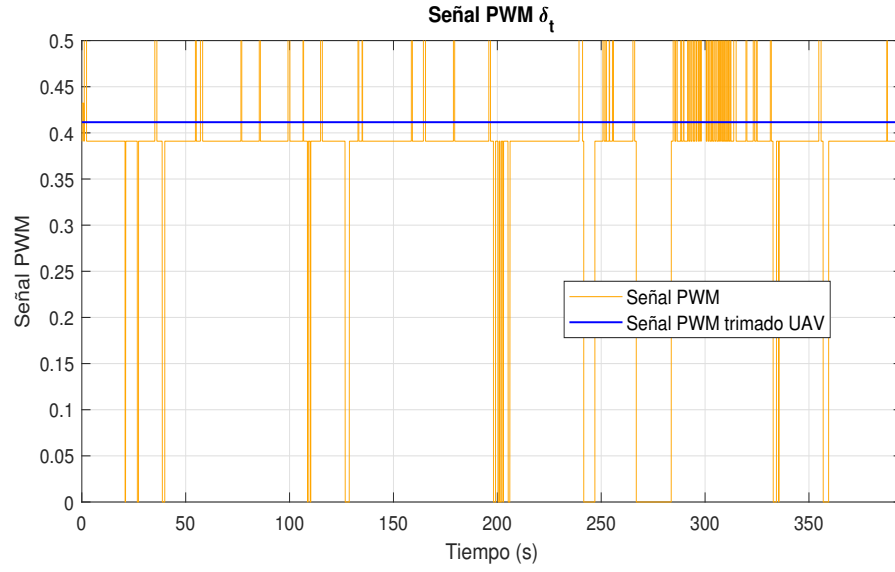


Figura 7.9 Señal PWM encargada de modelar el empuje..

observar cómo cuando el UAV entra en zona de descenso y ascenso toma los valores 0 y 1, respectivamente, tal y como se explicó en el Capítulo 3.1.5.

7.6 Elevones

En este apartado mostramos la deflexión de los elevones que actúan como alerones o elevadores del UAV. A partir de la ecuación (2.6) obtenemos la deflexión de los elevones derecho e izquierdo en función de la deflexión de alerones y elevadores obtenida en la simulación.

$$\delta_{er} = \frac{1}{2}(\delta_e - \delta_a) \quad (7.1a)$$

$$\delta_{el} = \frac{1}{2}(\delta_e + \delta_a) \quad (7.1b)$$

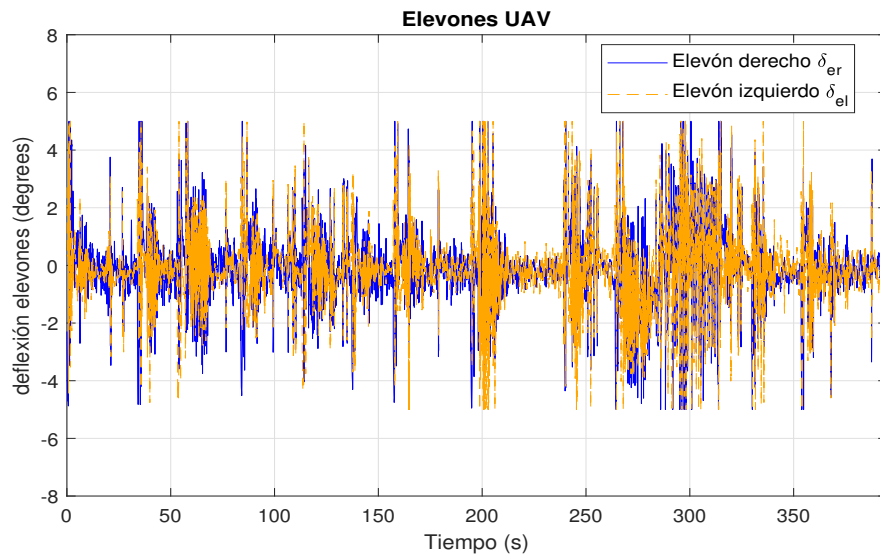


Figura 7.10 Deflexión de los elevones del UAV..

7.7 Coeficientes aerodinámicos

En este apartado mostramos la evolución del coeficiente de sustentación y resistencia del UAV. Comprobamos como el C_L es mucho mayor que el C_D

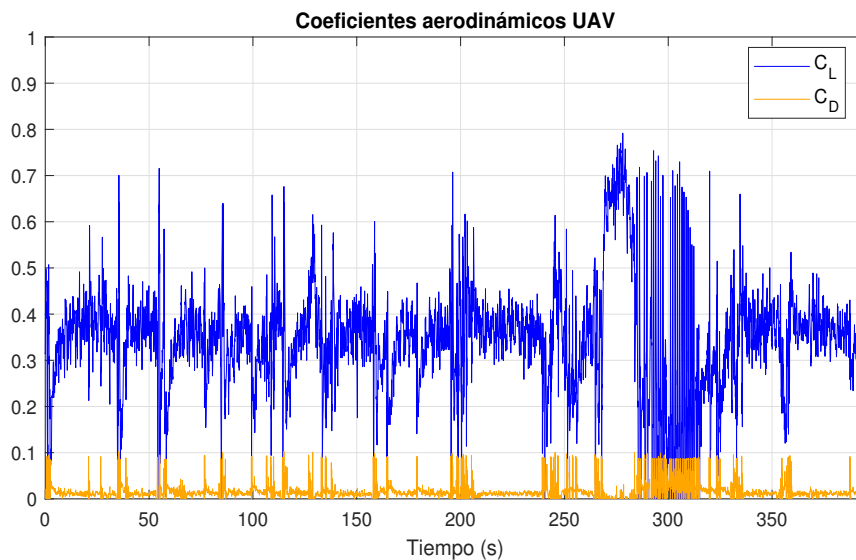


Figura 7.11 Evolución del C_L y C_D del UAV..

7.8 Acelerómetro

En este apartado mostramos las medidas tomadas por el acelerómetro.

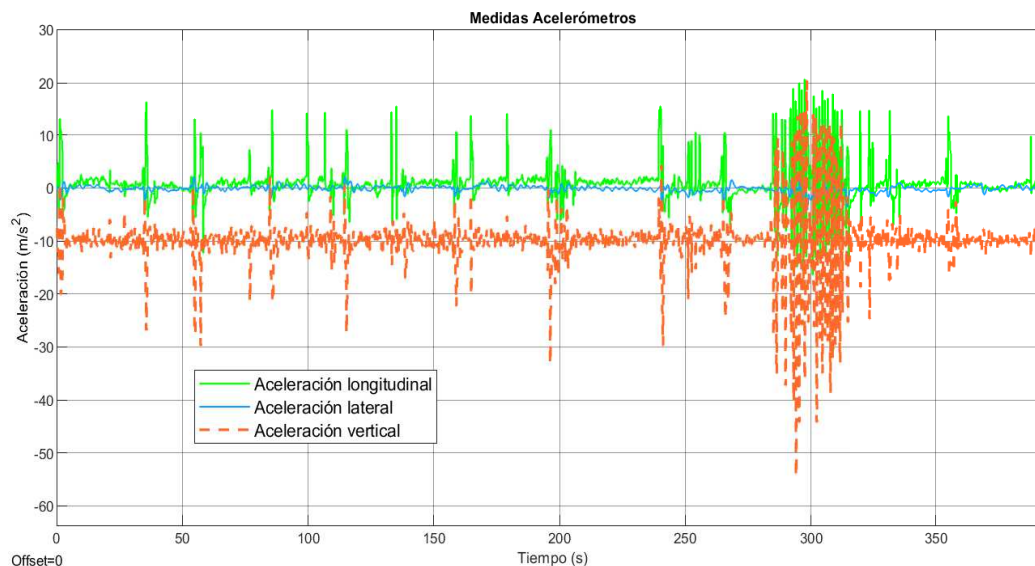


Figura 7.12 Medidas tomadas por el acelerómetro..

Observamos como las mayores variaciones se dan en las componentes longitudinal y vertical, algo totalmente lógico ya que la máquina de control de altitud se encuentra en todo momento controlando la Señal PWM del empuje y la deflexión de los elevadores. El valor negativo de la aceleración vertical se debe a que

durante la simulación el UAV se encuentra predominantemente en zona de retención de altitud o descenso, en la cual el empuje se hace nulo.

7.9 Giróscopo

En este apartado mostramos las medidas tomadas por el giróscopo. Vemos como las mayores variaciones de velocidad angular del UAV son aquellas en torno al eje longitudinal, provocado por el alabeo, y en torno al eje lateral, provocado por el cabeceo.

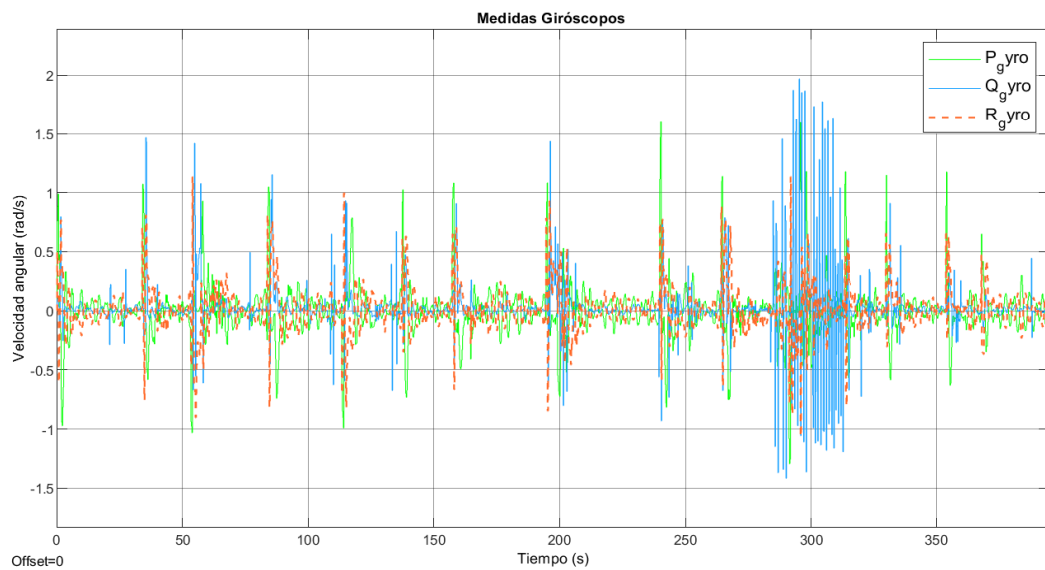


Figura 7.13 Medidas tomadas por el giróscopo..

Estos resultados son coherentes con el sistema de misión implementado en el UAV, el cual se encarga de calcular el valor de altitud de referencia necesaria para el controlador de altitud y el ángulo de guiñada necesario, el cual es controlado mediante el alabeo del UAV al no disponer de timón de profundidad.

7.10 Sistema GPS

En este apartado vamos a mostrar las medidas de velocidad y rumbo tomadas con el Sistema GPS. En cuanto a la velocidad, vamos a comparar el valor proporcionado por el GPS con la integración de las ecuaciones diferenciales del UAV.

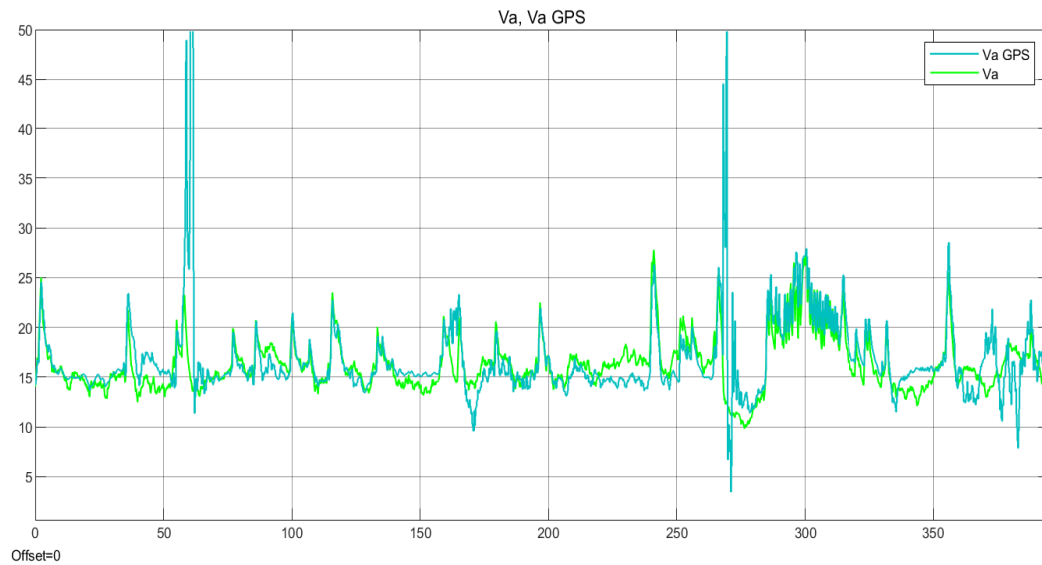


Figura 7.14 Comparación entre velocidad real y velocidad dada por GPS..

Observamos como los resultados ofrecidos por el Sistema GPS son razonadamente aceptables, salvo en instantes puntuales de tiempo en los cuales ofrece valores muy alejados del real. Estas pequeñas perturbaciones se pueden deber a la incertidumbre presente en las mediciones del Sistema GPS.

A continuación se muestra el rumbo obtenido mediante el GPS.

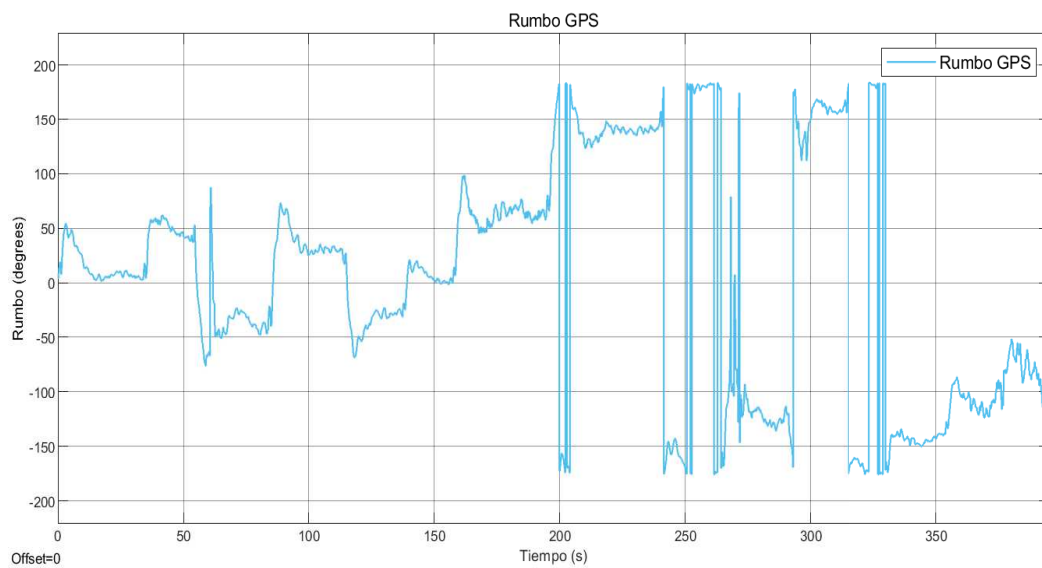


Figura 7.15 Rumbo del UAV dado por GPS..

8 Estudio de Robustez frente a Perturbaciones

Los resultados presentados en el Capítulo anterior correspondían a una simulación concreta: vuelo a 15 m/s con viento de 10 m/s y componente WSW (247.5°). Las perturbaciones de viento afectan enormemente a la actuación de una aeronave, y en el UAV de estudio de este proyecto estos efectos se multiplican al ser su masa inferior a 4 kg. Por esta razón en este Capítulo se va a llevar a cabo un estudio acerca de cómo influye el viento incidente en el UAV. En la Fig. 8.1 se muestra el viento registrado en el Aeropuerto de Sevilla durante el año 2018. La racha de viento mayor registrada durante el año corresponde a 25 m/s, mientras que la velocidad media corresponde a 10.3 m/s.

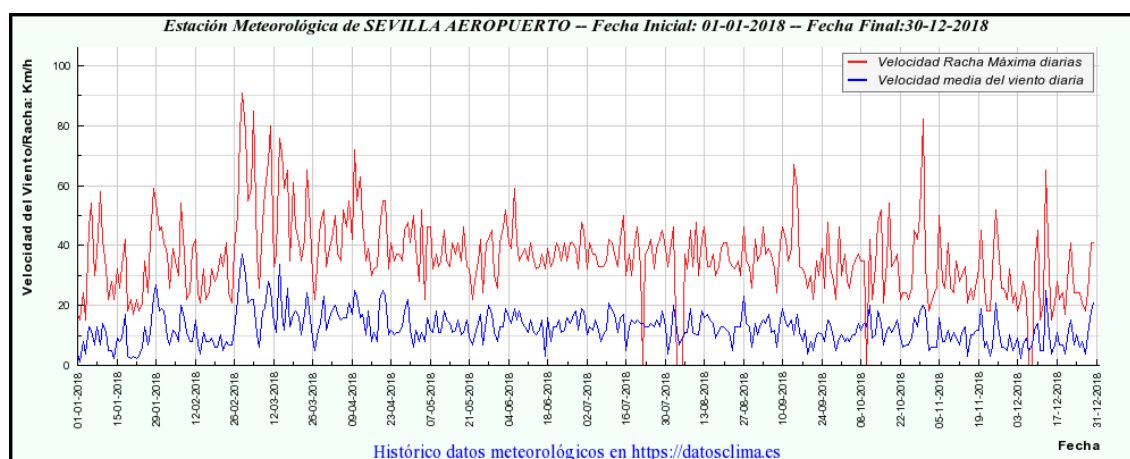


Figura 8.1 Viento registrado en el Aeropuerto de Sevilla en 2018..

8.1 Efecto de la intensidad del viento

En las imágenes (8.3) y (8.4) se muestra la trayectoria descrita por el UAV para vientos de distinta intensidad. Se comprueba como conforme aumenta la magnitud del viento más se ve afectada la actuación del UAV, hasta que llega un momento en el que no puede realizar la misión, con viento correspondiente a 25 m/s, aunque para vientos mayores de 15 m/s la actuación comienza a ser defectuosa.

Consideraremos el error cometido como la distancia entre el waypoint objetivo y el punto de la trayectoria en el que el sistema de misión considera que el UAV ha pasado por dicho waypoint. Se obtendrá el error cometido en cada punto de paso y se hallará el error medio cometido, así como el error máximo. También se calculará la media y la varianza de dichos valores. En la tabla 8.1 se presentan los resultados obtenidos.

A partir de los datos obtenidos para las simulaciones presentadas en la tabla anterior, se realiza una interpolación mediante spline cúbico para obtener la evolución del error cometido en función de la intensidad

Tabla 8.1 Error cometido frente a diferentes vientos..

Intensidad del viento (m/s)	Error máximo (m)	Error medio (m)	Varianza (m^2)
0	7.713	2.6719	3.9262
5	8.2054	4.0262	7.9507
8	12.221	5.6164	15.449
10	15.122	6.2597	22.563
13	18.695	8.0173	35.851
15	42.043	9.2618	100.04
17	36.767	10.077	93.041
20	34.867	12.763	89.07
22	87.247	20.616	579.47

del viento. Observamos cómo el error máximo cometido aumenta en mayor medida con la intensidad del viento que el error medio, ésto se debe al hecho de que en el segmento que une los waypoints 10 y 11 el UAV se encuentra el viento de cara, por lo que para poder llegar a realizar la maniobra tiene que pasar más lejos del waypoint 10.

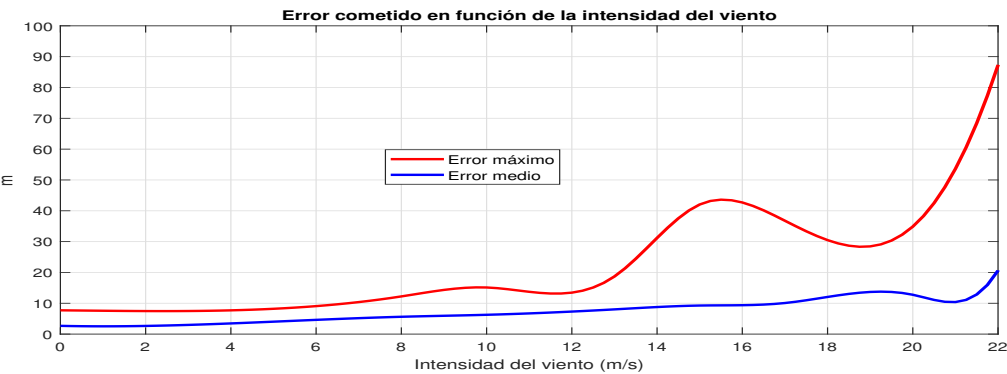


Figura 8.2 Error cometido en función de la intensidad del viento..

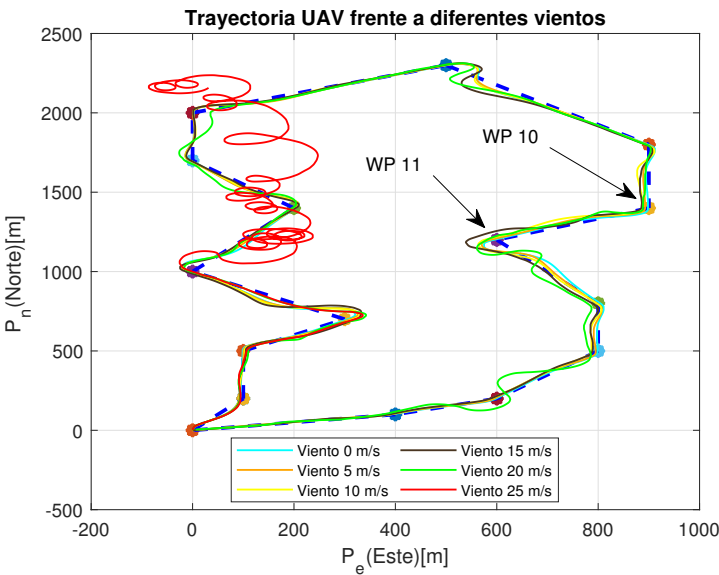


Figura 8.3 Trayectoria UAV para diferentes vientos..

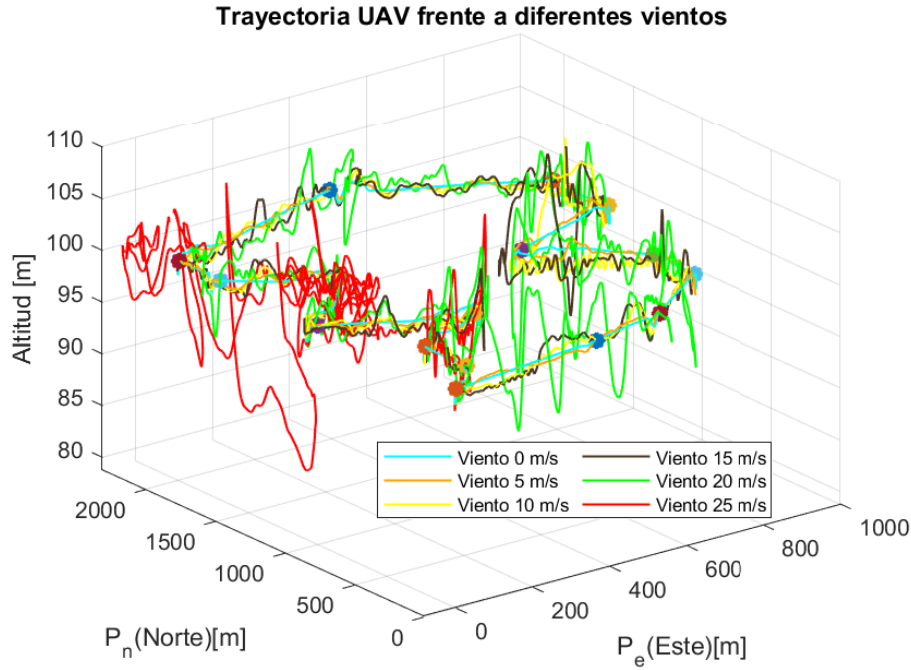


Figura 8.4 Trayectoria UAV para diferentes vientos..

8.2 Efecto de la dirección del viento

En este apartado vamos a estudiar el efecto de la dirección del viento en el comportamiento del UAV. Para ello realizaremos diversas simulaciones para un valor constante de intensidad del viento (10 m/s) e iremos variando la dirección del viento incidente. En la tabla 8.2 se presentan los resultados obtenidos.

Tabla 8.2 Error cometido frente a diferentes vientos..

Dirección del viento (degrees)	Waypoint error máximo	Error máximo (m)	Error medio (m)	Varianza (m^2)
0	12	12.456	5.5033	11.491
45	8	7.578	3.9328	4.9467
90	10	14.577	4.2084	12.826
135	10	11.113	3.5997	8.0059
180	12	9.5846	3.5683	8.5337
225	13	10.734	4.9675	13.022
270	12	27.282	8.9046	55.292
315	10	18.78	5.4207	22.82

Volviendo a realizar una interpolación mediante spline obtenemos la evolución del error cometido en función de la dirección del viento incidente. Observamos como el error decrece cuando el viento es de cola y aumenta cuando éste es de cara. Por tanto llegamos a la conclusión que interesa planificar la misión en función de la dirección predominante del viento en la zona a estudiar.

En la Fig. (8.6) se muestra la trayectoria descrita por el UAV en función de la dirección del viento incidente.

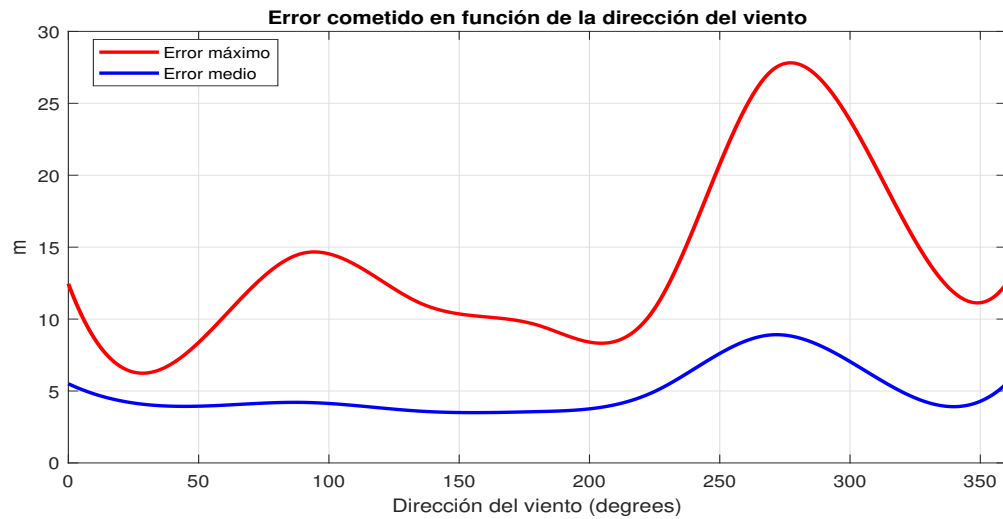


Figura 8.5 Error cometido en función de la dirección del viento..

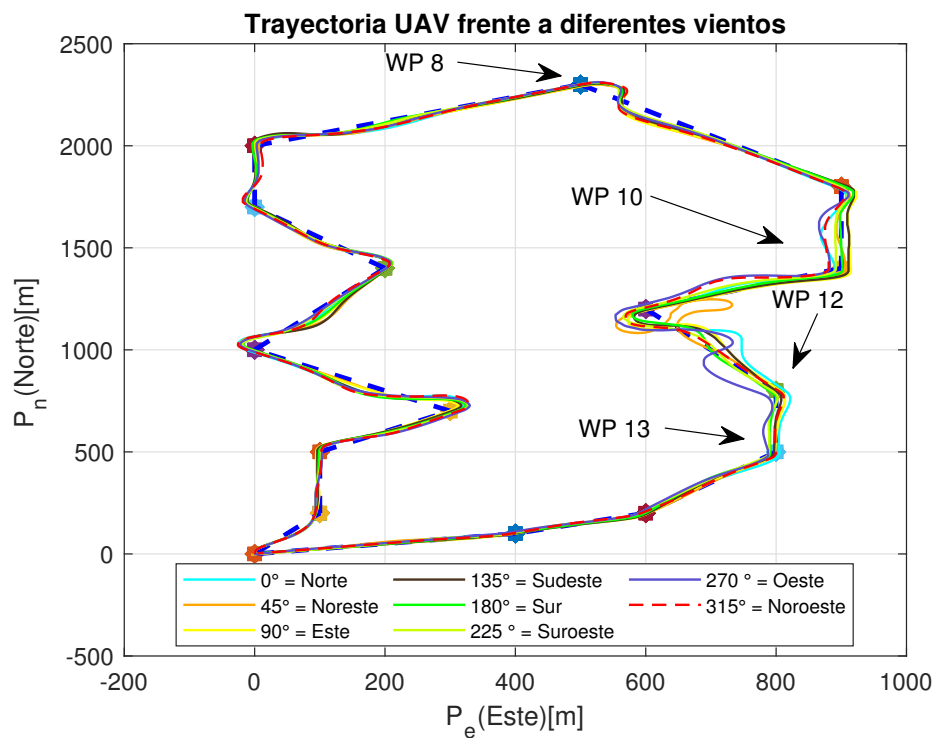


Figura 8.6 Trayectoria UAV para diferentes vientos..

8.3 Efecto de la velocidad de vuelo

En los apartados anteriores se ha estudiado el efecto del viento en el UAV, comprobándose como la intensidad de este afecta especialmente. Para vientos de intensidad 15 m/s el UAV es capaz de realizar la misión diseñada, pero requiere que el Autopiloto realice deflexiones de las superficies de control de manera muy brusca, algo que no es bueno para la estructura del UAV. Por ello, en este apartado se va estudiar el efecto de incrementar la velocidad de vuelo del UAV para hacer frente a vientos con intensidad considerable.

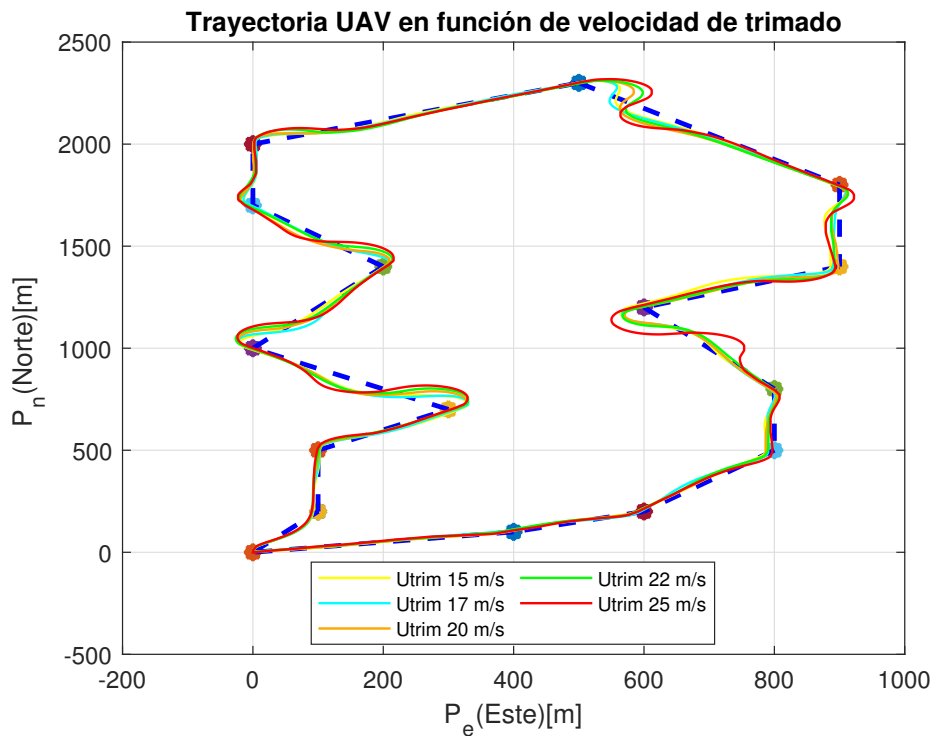
En la tabla 8.3 se muestran los resultados obtenidos para diferentes rachas de viento y velocidad de vuelo del UAV.

Tabla 8.3 Error cometido frente a diferentes vientos y velocidad de trimado..

Intensidad del viento (m/s)	Velocidad de trimado (m/s)	Error máximo (m)	Error medio (m)	Varianza (m^2)
5	17	8.6099	3.5131	6.1445
10	17	10.177	5.3421	11.919
15	17	15.276	7.8965	26.54
5	20	9.7684	3.7187	7.8495
10	20	12.739	4.8855	14.793
15	20	19.538	6.8215	38.632
20	20	27.113	7.7207	66.852
10	22	10.064	4.806	11.686
15	22	17.512	5.8693	23.151
20	22	34.96	8.1581	86.942
5	25	10.505	3.7746	10.719
10	25	10.343	4.2825	12.353
15	25	35.865	10.954	124.75

Al aumentar la velocidad de vuelo contrarrestamos los efectos del viento y se reduce considerablemente el tiempo de misión, pero si la ruta definida no es lo suficientemente extensa el error cometido puede aumentar debido a que el UAV no tiene tiempo para poder pasar tan cerca por todos los puntos como cuando se consideraba la misión a velocidades de vuelo más bajas, hasta llegar al extremo de que no es capaz de realizar la misión. Por ejemplo, para un viento de 5 m/s y velocidad de trimado de 25 m/s el UAV se queda estancado en un waypoint describiendo trayectorias circulares alrededor de él.

En la Fig. 8.7 se muestra una comparativa de la trayectoria descrita por el UAV para un viento de intensidad 10 m/s y componente WSW para diferentes velocidades de trimado del UAV.

**Figura 8.7** Trayectoria del UAV en función de la velocidad de trimado..

9 Implementación con Simulador de Vuelo FlightGear

La implementación con el simulador de vuelo FlightGear llevada a cabo durante este Capítulo es un extra opcional, pero que aporta una experiencia visual al objetivo de este Proyecto. FlightGear es un simulador de vuelo multiplataforma y de código libre que es compatible con Matlab-Simulink. A partir de los datos calculados durante la simulación el simulador es capaz de mostrar en tiempo real la misión llevada a cabo por la aeronave.

Para poder exportar los datos calculados durante la simulación en Simulink hacemos uso de dos bloques perteneciente al Aerospace Blockset de Simulink:

- **FlightGear Preconfigured 6DoF Animation:** bloque que permite exportar los valores de posición y actitud de la aeronave al simulador de vuelo de FlightGear. Los valores de posición deben proporcionarse en coordenadas geodésicas, por ejemplo el Sistema WGS84. Para poder configurar el bloque es necesario seleccionar la versión instalada de FlightGear ,introducir la dirección IP del ordenador donde se vaya a realizar la simulación, el puerto de destino (por defecto 5502), y el tiempo de muestro.
- **Run Generate Script:** se encarga de crear el fichero .bat para ser ejecutado por FlightGear. En este bloque es necesario especificar la aeronave con la cual se va realizar la simulación, que en nuestro caso ha sido *Malolo1*, un ala volante presente en la base de datos de FlightGear con características muy similares. También hay que introducir el aeropuerto en el que se va realizar la simulación (LEZL), la pista de despegue (27), altitud (328 ft) y rumbo (0°) iniciales, así como el offset de distancia (NM) y azimut($^\circ$).

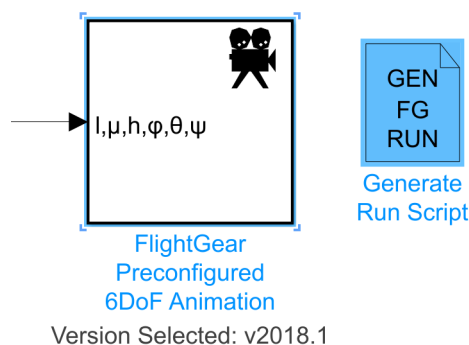


Figura 9.1 Bloques para ejecutar simulación en FlightGear..

A continuación se muestra la configuración introducida en Simulink de ambos bloques. Haciendo click en la ventana "Generate Script" creamos el fichero .bat, el cual al abrirlo nos cargará la situación configurada en la Fig. (9.2).

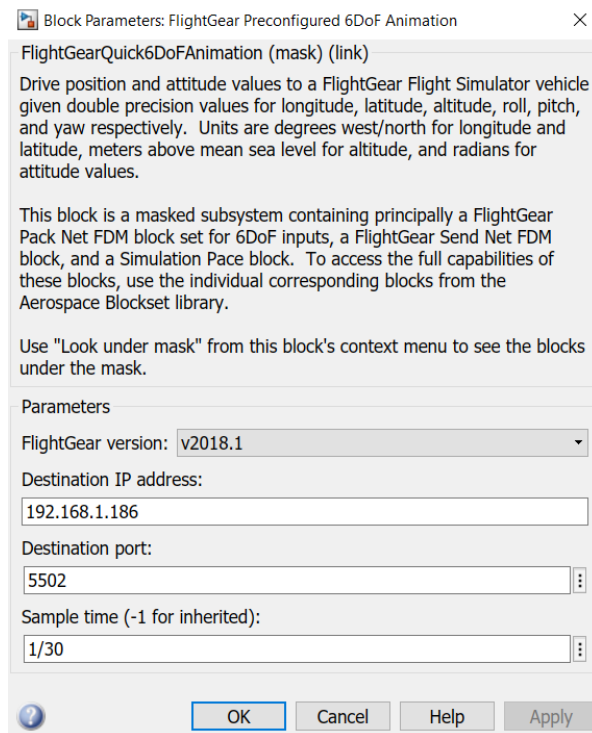


Figura 9.2 *Bloque de animación 6DoF de FlightGear.*

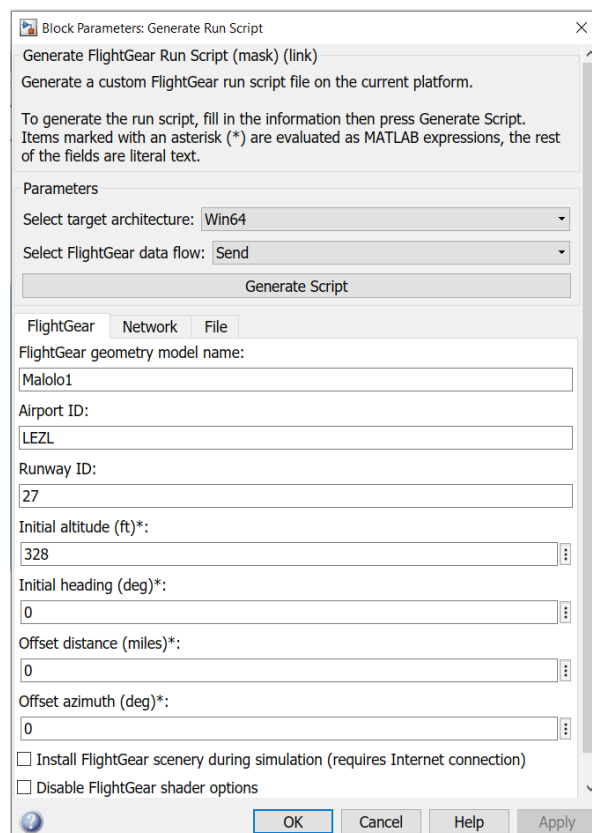


Figura 9.3 *Configuración del Bloque Run Generate Script.*

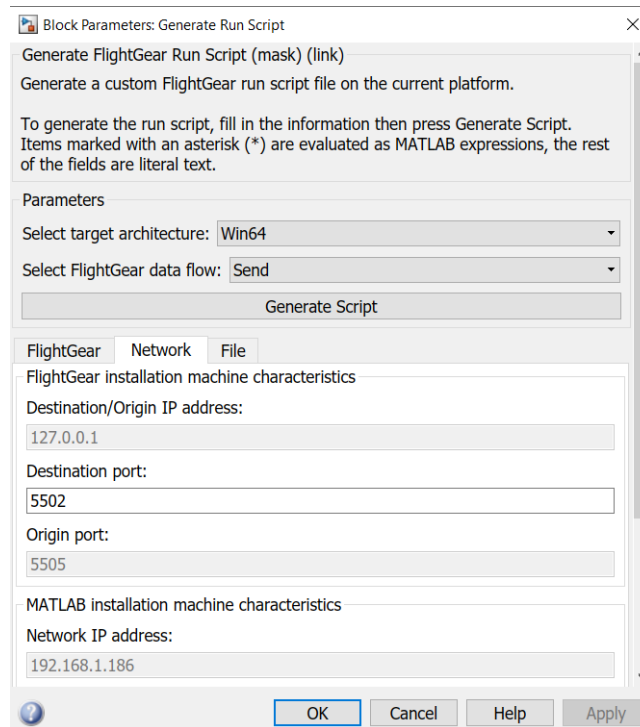


Figura 9.4 Configuración del Bloque Run Generate Script..

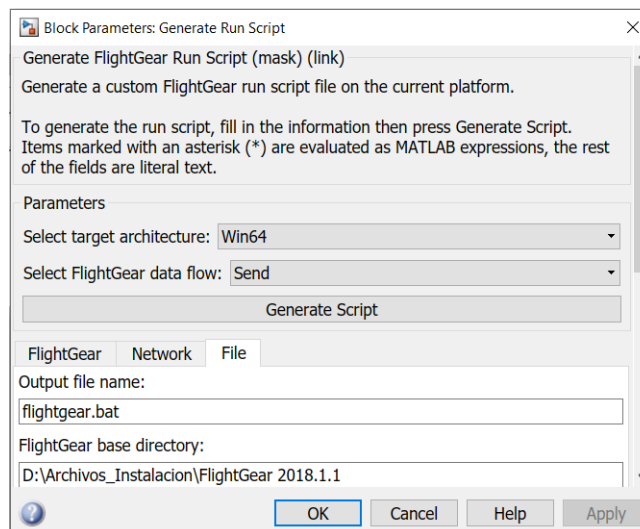


Figura 9.5 Configuración del Bloque Run Generate Script..

Así una vez configurados ambos bloques, abrimos el fichero .bat creado y comenzamos la simulación en Simulink. En la Fig. (9.6) se muestra cómo se vería nuestro UAV en FlightGear durante la simulación.

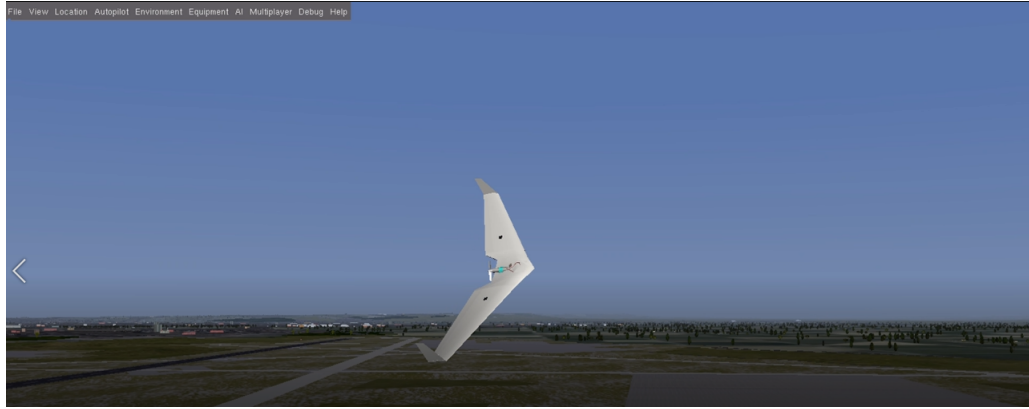


Figura 9.6 *Simulación en FlightGear..*

10 Conclusiones y Desarrollo Futuro

El objetivo de este proyecto era la de desarrollar un modelo matemático que describa el comportamiento dinámico del UAV Skywalker X8 y posteriormente estudiar diferentes misiones que pueda llevar a cabo. Una vez concluido podemos hacer una serie de valoraciones acerca de los resultados obtenidos:

- Al tratarse de un sistema dinámico no lineal el gasto computacional es elevado. Esto se traduce en la necesidad de emplear un equipo potente y el seleccionar una ruta no demasiado compleja. Particularmente, la simulación se ha realizado en un ordenador provisto con un procesador Intel Core i7-8750H 2.20GHz, memoria RAM de 8GB y tarjeta gráfica NVIDIA GeForce GTX 1060. Para llevar a cabo la simulación en FlightGear fue necesario calcular por una parte los parámetros necesarios para FlightGear y posteriormente introducir en un modelo Simulink distintos dichos parámetros desde el workspace de Matlab para poder realizar la simulación, ya que si se intentaba ejecutar en el mismo archivo la simulación de Simulink y la generación en FlightGear el ordenador proporcionaba resultados incorrectos.
- El sistema de misión implementado implica que el UAV pase por todos los waypoints definidos, lo cual puede dar lugar a que en el caso de que dos puntos se encuentren muy juntos tras un viraje el UAV deba salirse de la ruta óptima calculada por el algoritmo TSP. Una posible solución puede ser el diseñar el sistema de misión de manera que realice maniobras circulares, aunque no llegue a pasar exactamente por el punto en cuestión.
- El UAV al tener una masa de tan solo 4kg se ve afectado notablemente por la presencia de viento. A mayor valor de la intensidad del viento más perjudicado se ve el UAV, hasta llegar a un determinado momento en el que no es capaz de hacer frente al viento incidente. Para solucionar este problema es necesario aumentar la velocidad de vuelo para disminuir el efecto de dicho viento, pero tal y como se vio en el Estudio de Robustez en el Capítulo 8, aumentar la velocidad no siempre es posible, ya que puede introducir error en la trayectoria del UAV o que no sea capaz de realizar la misión. Por tanto, se concluye que a la hora de diseñar una misión hay que tener en cuenta una serie de factores muy importantes: previsión meteorológica, extensión de la zona a cubrir, elegir estratégicamente los waypoints de paso y la velocidad de vuelo del UAV. Teniendo en cuenta estas variables el UAV realiza la misión de manera satisfactoria.
- La simulación de un sistema tan complejo muestra cómo hay que tener cuidado a la hora de seleccionar todos y cada uno de los parámetros de diseño, ya que pequeños cambios en cualquier parámetro puede dar lugar a que el sistema no se comporte como se espera.

En cuanto a posibles líneas de desarrollo se pueden incluir las siguientes:

- Implementar un sistema de misión que sea capaz de detectar obstáculos y de realizar rutas tridimensionales.
- Sustituir los controladores PID implementados por leyes de control óptimo de trayectorias.
- Introducir diferentes modelos de perturbaciones atmosféricas para comprobar resultados.
- Comparar los resultados obtenidos en la simulación frente a datos de telemetría reales.

Anexos

A Representación en Google Eearth

En este anexo se muestra la trayectoria descrita por el UAV en las distintas simulaciones llevadas a cabo representadas en Google Earth.

- **Figura A.1:** Vuelo a 15 m/s con viento de intensidad 10 m/s y dirección WSW.
- **Figura A.2:** Trayectoria del UAV en función de la intensidad del viento
- **Figura A.3:** Trayectoria del UAV en función de la dirección del viento.
- **Figura A.4:** Trayectoria del UAV en función de la velocidad de trimado.

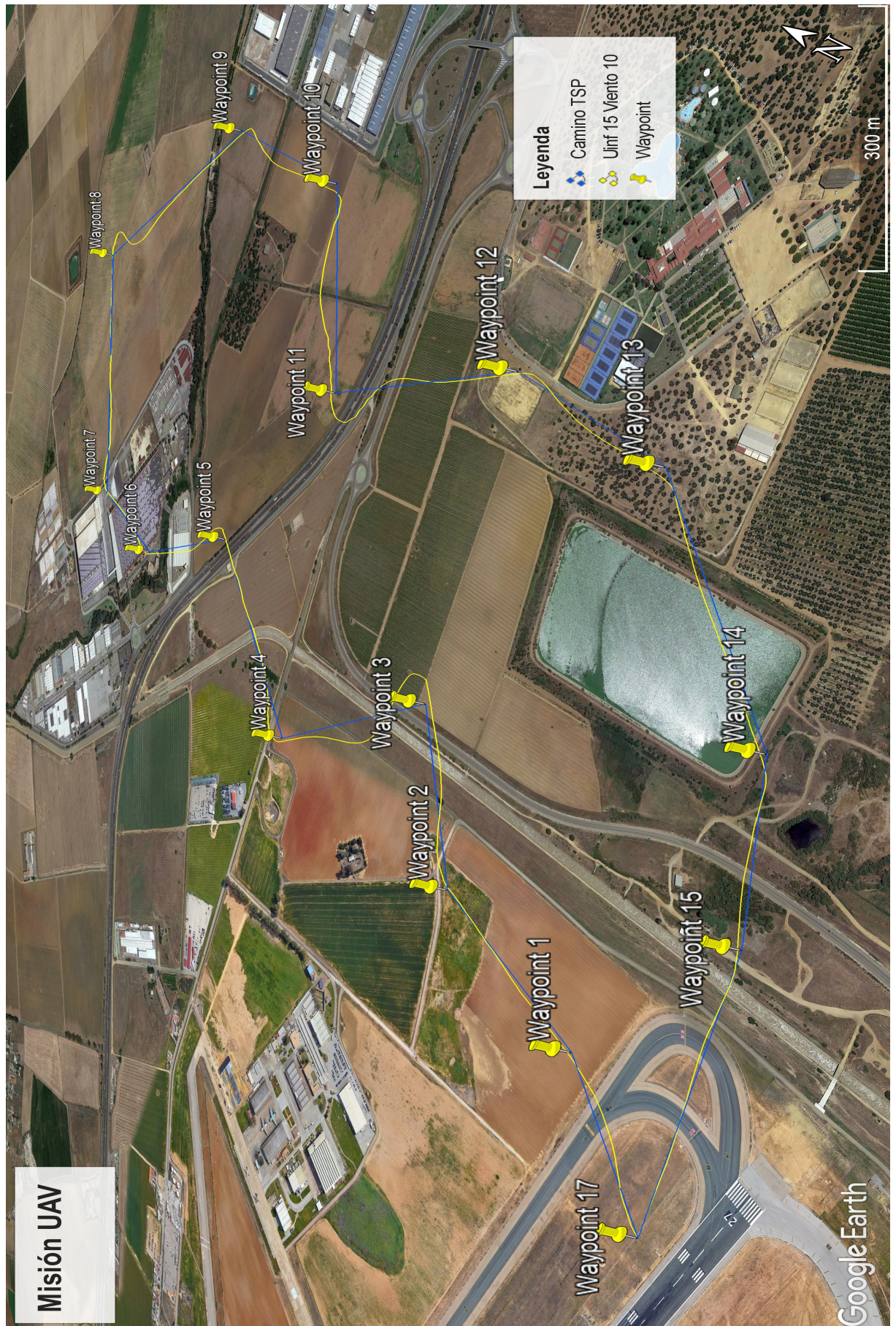


Figura A.1 Vuelo a 15 m/s con viento de intensidad 10 m/s y dirección WSW..

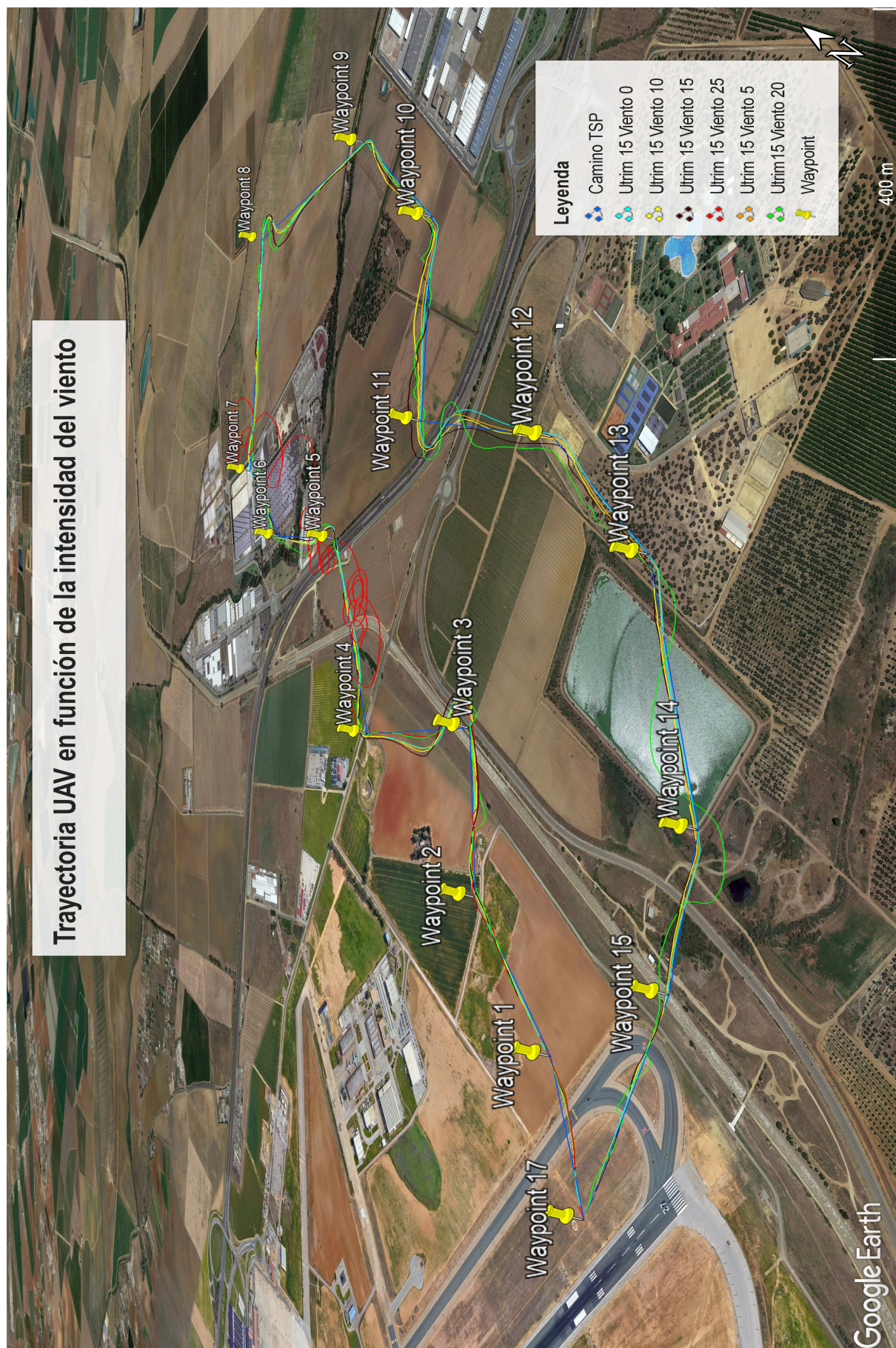


Figura A.2 Trayectoria del UAV en función de la intensidad del viento..

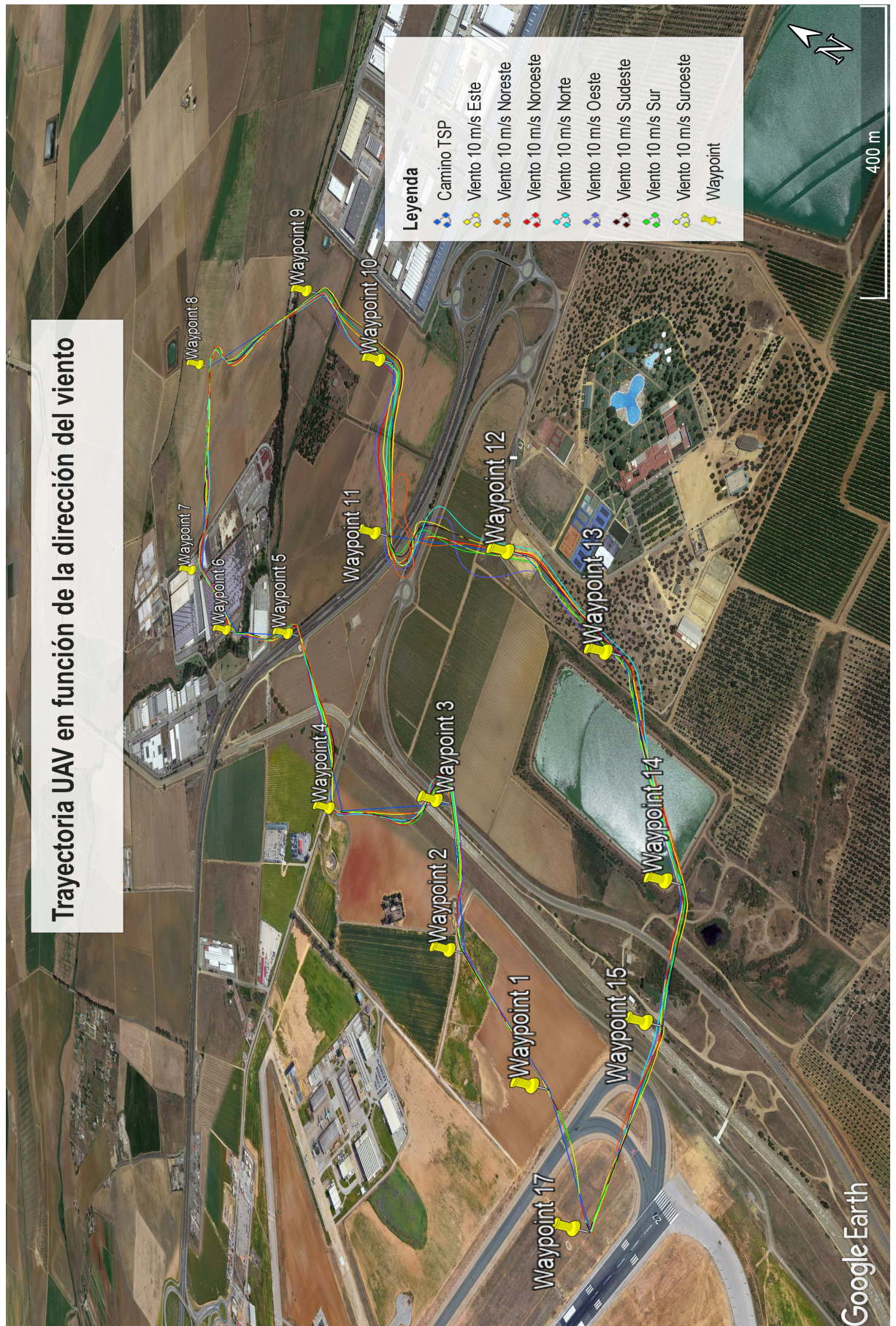


Figura A.3 Trayectoria del UAV en función de la dirección del viento..

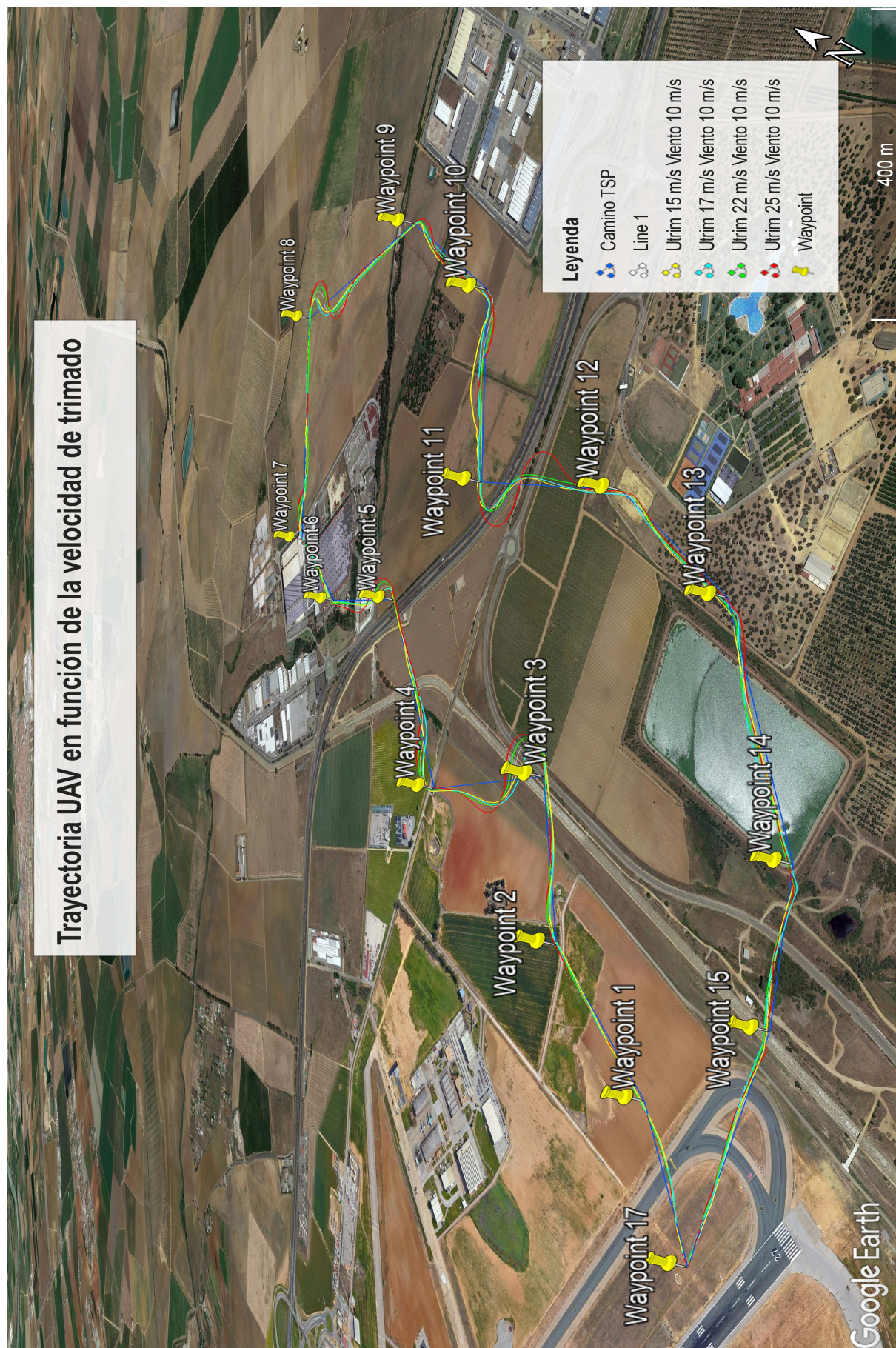


Figura A.4 Trayectoria del UAV en función de la dirección del viento..

Índice de Figuras

1.1	<i>X8 Skywalker.</i>	2
1.2	<i>Sistema de coordenadas de Ejes Tierra.</i>	2
1.3	<i>Sistema de Ejes de Navegación.</i>	3
1.4	<i>Sistema de Ejes Cuerpo.</i>	3
1.5	<i>Sistema de Ejes Viento.</i>	3
2.1	<i>Variables de estado.</i>	6
2.2	<i>Esferoide del sistema WGS84.</i>	8
2.3	<i>Fuerzas y momento aerodinámicos aplicadas en el centro aerodinámico del perfil.</i>	8
2.4	<i>Variables de control aerodinámicas en configuración estándar.</i>	9
2.5	<i>Variables de control aerodinámicas para cola en V.</i>	10
2.6	<i>Variables de control aerodinámicas del ala volante.</i>	10
2.7	<i>C_L frente a α.</i>	12
2.8	<i>C_D frente a α.</i>	12
3.1	<i>Regímenes de vuelo para el autopiloto longitudinal.</i>	19
3.2	<i>Diagrama de bloques del controlador de cabeceo.</i>	20
3.3	<i>Diagrama de bloques del controlador de altitud.</i>	21
3.4	<i>Diagrama de bloques del controlador de velocidad mediante cabeceo.</i>	22
3.5	<i>Diagrama de bloques del controlador de velocidad mediante empuje.</i>	23
3.6	<i>Máquina de estados para control longitudinal.</i>	24
3.7	<i>Diagrama de bloques del controlador de alabeo.</i>	25
3.8	<i>Diagrama de bloques del controlador de rumbo.</i>	26
4.1	<i>Altitud deseada para diseño de ruta longitudinal.</i>	28
4.2	<i>Algoritmo empleado para el cálculo de la ruta en línea recta.</i>	29
4.3	<i>Criterio de actualización de segmento de ruta..</i>	29
4.4	<i>Algoritmo de gestión de ruta.</i>	30
4.5	<i>Ejemplo del Problema TSP.</i>	31
4.6	<i>Algoritmo TSP.</i>	32
5.1	<i>Parámetros del modelo de error de Gauss-Markov.</i>	35
6.1	<i>Bloque Cinemática lineal en Simulink.</i>	39
6.2	<i>Bloque Cinemática angular en Simulink.</i>	39
6.3	<i>Bloque Dinámica lineal en Simulink.</i>	40
6.4	<i>Bloque Dinámica angular en Simulink.</i>	40
6.5	<i>Bloque WGS84 de Simulink.</i>	41
6.6	<i>Bloque Coeficientes Aerodinámicos en Simulink.</i>	41
6.7	<i>Bloque de Modelado del Viento en Simulink.</i>	41
6.8	<i>Bloque de Dryden de Simulink.</i>	42
6.9	<i>Viento predominante a 10 m de altitud.</i>	42

6.10	<i>Viento predominante a 80 m de altitud.</i>	42
6.11	<i>Bloque de Fuerzas en Simulink.</i>	43
6.12	<i>Bloque de Momentos en Simulink.</i>	43
6.13	<i>Bloque de controladores en Simulink.</i>	44
6.14	<i>Máquina de estados de control longitudinal en Simulink.</i>	44
6.15	<i>Controlador de Cabeceo en Simulink.</i>	44
6.16	<i>Controlador de Altitud en Simulink.</i>	45
6.17	<i>Controlador de Velocidad mediante Cabeceo en Simulink.</i>	45
6.18	<i>Controlador de Velocidad mediante Empuje en Simulink.</i>	45
6.19	<i>Controlador de Alabeo en Simulink.</i>	46
6.20	<i>Controlador de Rumbo en Simulink.</i>	46
6.21	<i>Sistema de misión en Simulink.</i>	46
6.22	<i>Acelerómetro en Simulink.</i>	47
6.23	<i>Giróscopo en Simulink.</i>	47
6.24	<i>Altímetro barométrico en Simulink.</i>	48
6.25	<i>Anemómetro en Simulink.</i>	48
6.26	<i>Magnetómetro en Simulink.</i>	48
6.27	<i>Sistema GPS en Simulink.</i>	49
6.28	<i>EKF de Posición en Simulink.</i>	49
6.29	<i>EKF de Actitud en Simulink.</i>	50
7.1	<i>Misión realizada por el UAV en Matlab-Simulink.</i>	52
7.2	<i>Posición Norte del UAV.</i>	52
7.3	<i>Posición Este del UAV.</i>	53
7.4	<i>Altitud del UAV.</i>	53
7.5	<i>Alabeo del UAV.</i>	54
7.6	<i>Cabeceo del UAV.</i>	54
7.7	<i>Guiñada del UAV.</i>	55
7.8	<i>Velocidad del UAV.</i>	55
7.9	<i>Señal PWM encargada de modelar el empuje.</i>	56
7.10	<i>Deflexión de los elevones del UAV.</i>	56
7.11	<i>Evolución del C_L y C_D del UAV.</i>	57
7.12	<i>Medidas tomadas por el acelerómetro.</i>	57
7.13	<i>Medidas tomadas por el giróscopo.</i>	58
7.14	<i>Comparación entre velocidad real y velocidad dada por GPS.</i>	59
7.15	<i>Rumbo del UAV dado por GPS.</i>	59
8.1	<i>Viento registrado en el Aeropuerto de Sevilla en 2018.</i>	61
8.2	<i>Error cometido en función de la intensidad del viento.</i>	62
8.3	<i>Trayectoria UAV para diferentes vientos.</i>	62
8.4	<i>Trayectoria UAV para diferentes vientos.</i>	63
8.5	<i>Error cometido en función de la dirección del viento.</i>	64
8.6	<i>Trayectoria UAV para diferentes vientos.</i>	64
8.7	<i>Trayectoria del UAV en función de la velocidad de trimado.</i>	65
9.1	<i>Bloques para ejecutar simulación en FlightGear.</i>	67
9.2	<i>Bloque de animación 6DoF de FlightGear.</i>	68
9.3	<i>Configuración del Bloque Run Generate Script.</i>	68
9.4	<i>Configuración del Bloque Run Generate Script.</i>	69
9.5	<i>Configuración del Bloque Run Generate Script.</i>	69
9.6	<i>Simulación en FlightGear.</i>	70
A.1	<i>Vuelo a 15 m/s con viento de intensidad 10 m/s y dirección WSW.</i>	76
A.2	<i>Trayectoria del UAV en función de la intensidad del viento.</i>	77
A.3	<i>Trayectoria del UAV en función de la dirección del viento.</i>	78
A.4	<i>Trayectoria del UAV en función de la dirección del viento.</i>	79

Índice de Tablas

2.1	<i>Variables de estado.</i>	5
2.2	<i>Parámetros UAV Skywalker X8.</i>	14
2.3	<i>Parámetros de ráfaga del modelo Dryden [6]</i>	16
8.1	<i>Error cometido frente a diferentes vientos.</i>	62
8.2	<i>Error cometido frente a diferentes vientos.</i>	63
8.3	<i>Error cometido frente a diferentes vientos y velocidad de trimado.</i>	65

Bibliografía

- [1] Gryte, K. *High Angle of Attack Landing of an Unmanned Aerial Vehicle*. MSc Tesis. Norwegian University of Science and Technology. Trondheim, Norway. 2015.
- [2] Randal W. Beard, Timothy W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press. Princeton and Oxford.
- [3] Rísquez Ruiz, A. *Modelado y simulación del Skywalker X8*. 2017.
- [4] R. F. Stengel. *Flight Dynamics*. Princeton, NJ: Princeton University Press. 2004.
- [5] T. R. Yechout, S. L. Morris, D. E. Bossert, W. F. Hallgren. *Introduction to Aircraft Flight Mechanics*. AIAA Education Series, American Institute of Aeronautics and Astronautics. 2003.
- [6] J. W. Langelaan, N. Alley, and J. Niedhoefer. *Wind field estimation for small unmanned aerial vehicles*. AIAA Guidance, Navigation, and Control Conference, Toronto, Canada. 2010.
- [7] MathWorks. *Traveling Salesman Problem: Solver-Based*. <https://es.mathworks.com/help/optim/ug/travelling-salesman-problem.html?lang=en>
- [8] Fernández Camacho, E. *Apuntes de la asignatura "Sistemas de control y guiado"*. Universidad de Sevilla. Sevilla, España. 2016.
- [9] Limón Marruedo, D. *Apuntes de la asignatura "Sistemas de control y guiado"*. Universidad de Sevilla. Sevilla, España. 2016.
- [10] Gavilán Jiménez, F. *Apuntes de la asignatura "Mecánica del Vuelo Avanzada"*. Universidad de Sevilla. Sevilla, España. 2017.
- [11] Bolzern, Paolo ; Scattolini, Riccardo, coaut. *Fundamentos de Control Automático*. 2009.
- [12] Página web de FlightGear. <https://www.flightgear.org/>
- [13] Enlace de descarga del modelo de aeronave para FlightGear. http://ftp.igh.cnrs.fr/pub/flightgear/ftp/Aircraft-2.0.0/Malolo1_0.0.zip
- [14] Enlace de descarga del escenario utilizado en FlightGear. <http://ns334561.ip-5-196-65.eu/~fgscenery/WS2.0/scenery-2.0.1.html>
- [15] MathWorks. *FlightGear Preconfigured 6DoF Animation*. <https://es.mathworks.com/help/aeroblks/flightgearpreconfigured6dofanimation.html>